

LPIC-1

Versión 5.0
Español

102

Table of Contents

TEMA 105: SHELLS Y SCRIPTS **1**
105.1 Personalizar y usar el entorno de shell **2**
105.1 Lección 1 4
Introducción 4
Tipos de shell: Interactivo vs. No Interactivo y Inicio de sesión vs Sin inicio de sesión 5
Ejercicios guiados 19
Ejercicios de exploración 21
Resumen 23
Respuestas a los ejercicios guiados 25
Respuestas a los ejercicios de exploración 27
105.1 Lección 2 29
Introducción 29
Variables: Asignación y referencia 29
Variables locales o de Shell 34
Variables globales o de entorno 36
Ejercicios guiados 47
Ejercicios de exploración 50
Resumen 52
Respuestas a los ejercicios guiados 54
Respuestas a los ejercicios de exploración 58
105.1 Lección 3 60
Introducción 60
Creando Alias 60
Creando funciones 65
Ejercicios guiados 76
Ejercicios de exploración 80
Resumen 81
Respuesta a los ejercicios guiados 83
Respuestas a los ejercicios de exploración 88
105.2 Personalizar y escribir scripts sencillos **89**
105.2 Lección 1 91
Introducción 91
Estructura y ejecución del script 92
Variables 94
Expresiones aritméticas 98
Ejecución condicional 98
Salidas de un Script 99
Ejercicios guiados 102

Ejercicios de exploración . . . 103
Resumen . . . 104
Respuesta a los ejercicios guiados . . . 105
Respuestas a los ejercicios de exploración . . . 106
105.2 Lección 2 . . . 107
Introducción . . . 107
Pruebas ampliadas . . . 107
Construcciones de bucle . . . 113
Un ejemplo más elaborado . . . 115
Ejercicios guiados . . . 120
Ejercicios de exploración . . . 122
Resumen . . . 123
Respuesta a los ejercicios guiados . . . 124
Respuestas a los ejercicios de exploración . . . 126
TEMA 106: INTERFACES DE USUARIO Y ESCRITORIOS . . . 127
106.1 Instalar y configurar X11 . . . 128
106.1 Lección 1 . . . 129
Introducción . . . 129
Arquitectura de sistema X Window . . . 130
Configuración de un servidor X . . . 133
Wayland . . . 138
Guided Exercises . . . 140
Explorational Exercises . . . 141
Resumen . . . 142
Respuesta a los ejercicios guiados . . . 143
Respuestas a los ejercicios de exploración . . . 144
106.2 Escritorios gráficos . . . 145
106.2 Lección 1 . . . 146
Introducción . . . 146
Sistema X Window . . . 147
Ambientes de escritorio . . . 147
Entornos de escritorio populares . . . 149
Interoperabilidad de escritorio . . . 151
Acceso no local . . . 152
Guided Exercises . . . 155
Explorational Exercises . . . 156
Resumen . . . 157
Respuesta a los ejercicios guiados . . . 158
Respuestas a los ejercicios de exploración . . . 159
106.3 Accesibilidad . . . 160

106.3 Lección 1 161
Introducción 161
Configuración de accesibilidad 161
Asistencia de teclado y mouse 162
Deficiencias visuales 164
Guided Exercises 166
Explorational Exercises 167
Resumen 168
Respuesta a los ejercicios guiados 169
Respuestas a los ejercicios de exploración 170
TEMA 107: TAREAS ADMINISTRATIVAS 171
107.1 Administrar cuentas de usuario y de grupo y los archivos de sistema relacionados con ellas 172
107.1 Lección 1 174
Introducción 174
Agregando cuentas de usuario 174
Modificación de las cuentas de usuario 176
Eliminando cuentas de usuario 178
El directorio skel 179
El archivo `/etc/login.defs` 179
El comando `passwd` 180
El comando `chage` 182
Ejercicios guiados 183
Ejercicios de exploración 184
Resumen 185
Respuesta a los ejercicios guiados 187
Respuestas a los ejercicios de exploración 189
107.1 Lección 2 192
Introducción 192
/etc/passwd 193
/etc/group 194
/etc/shadow 194
/etc/gshadow 195
Filtrar las bases de datos de contraseñas y grupos 195
Ejercicios guiados 197
Ejercicios de exploración 199
Resumen 200
Respuesta a los ejercicios guiados 201
Respuestas a los ejercicios de exploración 203
107.2 Automatizar tareas administrativas del sistema mediante la programación de

trabajos **205**
107.2 Lección 1 207
Introducción 207
Programar trabajos con Cron 207
Crontabs de usuario 208
Crontabs de sistema 209
Especificaciones de tiempo particulares 210
Variables de Crontab 210
Crear trabajos en un cron de usuario 211
Crear crones de sistema 212
Configurar el acceso a la programación de tareas 213
Una alternativa a cron 213
Ejercicios guiados 216
Ejercicios de exploración 218
Resumen 219
Respuestas a los ejercicios guiados 220
Respuestas a los ejercicios de exploración 222
107.2 Lección 2 224
Introducción 224
Programar tareas con `at` 224
Listar tareas programadas con `atq` 225
Borrar tareas con `atrm` 226
Configurar el acceso a la programación de tareas 226
Especificaciones de tiempo 227
Una alternativa a `at` 227
Ejercicios guiados 229
Ejercicios de exploración 230
Resumen 231
Respuestas a los ejercicios guiados 232
Respuestas a los ejercicios de exploración 233
107.3 Localización e internacionalización **235**
107.3 Lección 1 237
Introducción 237
Zonas horarias 238
Horario de verano (Daylight Saving Time) 243
Lenguaje y codificación de caracteres 243
Conversión de la codificación 247
Ejercicios guiados 248
Ejercicios de exploración 249
Resumen 250

Respuestas a los ejercicios guiados . . . 251
Respuestas a los ejercicios de exploración . . . 252
TEMA 108: SERVICIOS ESENCIALES DEL SISTEMA . . . 253
108.1 Mantener la hora del sistema . . . 254
108.1 Lección 1 . . . 256
Introducción . . . 256
Date . . . 257
Reloj de hardware . . . 259
timedatectl . . . 260
Establecer la zona horaria sin timedatectl . . . 262
Establecer la fecha y la hora sin timedatectl . . . 263
Ejercicios guiados . . . 265
Ejercicios de exploración . . . 267
Resumen . . . 268
Respuesta a los ejercicios guiados . . . 270
Respuestas a los ejercicios de exploración . . . 272
108.1 Lección 2 . . . 273
Introducción . . . 273
timedatectl . . . 275
NTP Daemon . . . 276
Configuración NTP . . . 277
pool.ntp.org . . . 278
ntpdate . . . 278
ntpq . . . 279
chrony . . . 280
Ejercicios guiados . . . 284
Ejercicios de exploración . . . 286
Resumen . . . 287
Respuesta a los ejercicios guiados . . . 288
Respuestas a los ejercicios de exploración . . . 290
108.2 Registros del sistema . . . 291
108.2 Lección 1 . . . 293
Introducción . . . 293
Registro del sistema . . . 293
Ejercicios guiados . . . 316
Ejercicios de exploración . . . 318
Resumen . . . 319
Respuesta a los ejercicios guiados . . . 320
Respuestas a los ejercicios de exploración . . . 323
108.2 Lección 2 . . . 324

Introducción 324
Fundamentos de `systemd` 324
The System Journal: `systemd-journald` 325
Ejercicios guiados 345
Ejercicios de exploración 348
Resumen 349
Respuestas a los ejercicios guiados 351
Respuestas a los ejercicios de exploración 354
108.3 Conceptos básicos del Agente de Transferencia de Correo 355
108.3 Lección 1 356
Introducción 356
MTA local y remoto 357
MTAs de Linux 358
El comando `mail` y los agentes de usuario de correo (MUA) 364
Personalización de la entrega 365
Ejercicios guiados 367
Ejercicios de exploración 368
Resumen 369
Respuestas a los ejercicios guiados 370
Respuestas a los ejercicios de exploración 371
108.4 Gestión de la impresión y de las impresoras 372
108.4 Lección 1 373
Introducción 373
El servicio CUPS 374
Instalación de una impresora 378
Gestión de impresoras 381
Envío de trabajos de impresión 382
Gestión de los trabajos de impresión 385
Eliminación de impresoras 386
Ejercicios guiados 388
Ejercicios de exploración 389
Resumen 390
Respuestas a los ejercicios guiados 392
Respuestas a los ejercicios de exploración 393
TEMA 109: FUNDAMENTOS DE REDES 395
109.1 Fundamentos de los protocolos de Internet 396
109.1 Lección 1 397
Introducción 397
IP (Internet Protocol) 397
Ejercicios guiados 406

Ejercicios de exploración 407
Resumen 408
Respuestas a los ejercicios guiados 409
Respuestas a los ejercicios de exploración 410
109.1 Lección 2 411
Introducción 411
Protocolo de Control de Transmisión (TCP) 413
Protocolo de Datagramas de Usuario (UDP) 413
Protocolo de mensajes de control de Internet (ICMP) 413
IPv6 413
Ejercicios guiados 417
Ejercicios de exploración 418
Resumen 419
Respuestas a los ejercicios guiados 420
Respuestas a los ejercicios de exploración 421
109.2 Configuración de red persistente 422
109.2 Lección 1 423
Introducción 423
La interfaz de red 423
Nombres de interfaces 425
Gestión de interfaces 426
Nombres locales y remotos 428
Ejercicios guiados 433
Ejercicios de exploración 434
Resumen 435
Respuestas a los ejercicios guiados 436
Respuestas a los ejercicios de exploración 437
109.2 Lección 2 438
Introducción 438
systemd-networkd 443
Ejercicios guiados 446
Ejercicios de exploración 447
Resumen 448
Respuestas a los ejercicios guiados 449
Respuestas a los ejercicios de exploración 450
109.3 Resolución de problemas básicos de red 451
109.3 Lección 1 453
Introducción 453
Sobre el comando `ip` 454
Revisión de máscaras de red y enrutamiento 455

Configurar una interfaz . . . 456
La tabla de enrutamiento . . . 458
Ejercicios guiados . . . 462
Ejercicios de exploración . . . 463
Resumen . . . 464
Respuestas a los ejercicios guiados . . . 465
Respuestas a los ejercicios de exploración . . . 467
109.3 Lección 2 . . . 469
Introducción . . . 469
Probar las conexiones con `ping` . . . 469
Traceroute . . . 470
Búsqueda de MTU con `tracepath` . . . 473
Crear conexiones arbitrarias . . . 474
Ver conexiones y oyentes actuales . . . 475
Ejercicios guiados . . . 478
Ejercicios de exploración . . . 479
Resumen . . . 480
Respuestas a los ejercicios guiados . . . 482
Respuestas a los ejercicios de exploración . . . 484
109.4 Configuración DNS en el lado del cliente . . . 486
109.4 Lección 1 . . . 487
Introducción . . . 487
Proceso de resolución de nombres . . . 487
Clases de DNS . . . 488
Herramientas de resolución de nombres . . . 491
Ejercicios guiados . . . 498
Ejercicios de exploración . . . 499
Resumen . . . 500
Respuesta a los ejercicios guiados . . . 501
Respuestas a los ejercicios de exploración . . . 502
TEMA 110: SEGURIDAD . . . 504
110.1 Tareas de administración de seguridad . . . 505
110.1 Lección 1 . . . 507
Introducción . . . 507
Gestión y caducidad de contraseñas . . . 510
Descubrir los puertos abiertos . . . 514
Límites en los inicios de sesión de los usuarios, los procesos y el uso de la memoria . . . 521
Tratar con usuarios registrados . . . 523
Configuración y uso básico de `sudo` . . . 526
Ejercicios guiados . . . 532

Ejercicios de exploración . . . 536
Resumen . . . 537
Respuesta a los ejercicios guiados . . . 539
Respuestas a los ejercicios de exploración . . . 543
110.2 Configuración de la seguridad del sistema . . . 544
110.2 Lección 1 . . . 545
Introducción . . . 545
Mejorar la seguridad de la autenticación con shadow password . . . 545
Cómo utilizar un superdemonio para escuchar las conexiones de red entrantes . . . 547
Comprobación de servicios en busca de daemons innecesarios . . . 553
TCP Wrappers como una especie de Firewall simple . . . 554
Ejercicios guiados . . . 556
Ejercicios de exploración . . . 557
Resumen . . . 558
Respuestas a los ejercicios guiados . . . 560
Respuestas a los ejercicios de exploración . . . 561
110.3 Protección de datos mediante cifrado . . . 562
110.3 Lección 1 . . . 564
Introducción . . . 564
Configuración y uso básico del cliente OpenSSH . . . 565
El papel de las claves del servidor OpenSSH . . . 571
Túneles de puertos SSH . . . 573
Ejercicios guiados . . . 578
Ejercicios de exploración . . . 580
Resumen . . . 581
Respuestas a los ejercicios guiados . . . 582
Respuestas a los ejercicios de exploración . . . 585
110.3 Lección 2 . . . 586
Introducción . . . 586
Configuración básica de GnuPG, uso y revocación . . . 586
Usar GPG para cifrar, descifrar, firmar y verificar archivos . . . 593
Ejercicios guiados . . . 598
Ejercicios de exploración . . . 600
Resumen . . . 601
Respuestas a los ejercicios guiados . . . 602
Respuestas a los ejercicios de exploración . . . 604
Pie de imprenta . . . 605



Tema 105: Shells y scripts



105.1 Personalizar y usar el entorno de shell

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 105.1

Importancia

4

Áreas de conocimiento clave

- Establecer variables de entorno (e.g. `PATH`) al inicio de sesión o al generar un nuevo shell.
- Escribir funciones en Bash para secuencias de comandos usadas con frecuencia.
- Mantener el esqueleto de directorios para nuevas cuentas de usuario.
- Establecer el directorio adecuado en la ruta de búsqueda de comandos.

Lista parcial de archivos, términos y utilidades

- `.`
- `source`
- `/etc/bash.bashrc`
- `/etc/profile`
- `env`
- `export`
- `set`
- `unset`
- `~/.bash_profile`

- `~/.bash_login`
- `~/.profile`
- `~/.bashrc`
- `~/.bash_logout`
- `function`
- `alias`



105.1 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	105 Shells y scripts
Objetivo:	105.1 Personalizar y usar el entorno de shell
Lección:	3 de 3

Introducción

El shell es posiblemente la herramienta más poderosa en un sistema operativo Linux y puede definirse como una interfaz entre el usuario y el kernel. Tiene la función de Interpretar los comandos introducidos por el usuario, por lo tanto, los administradores de sistemas deben ser hábiles en su uso. Como probablemente sabemos, el Bourne Again Shell (*Bash*) es el shell *de facto* de la gran mayoría de las distribuciones de Linux.

El momento que el sistema operativo inicia, lo primero que el Bash (o cualquier otro shell) realiza, es ejecutar una serie de scripts de inicio. Estos scripts personalizan el entorno de sesión. Existen varios scripts para todo el sistema operativo, asi también para usuarios específicos. En estos scripts podemos elegir las preferencias o configuraciones que mejor se adapten a las necesidades de nuestros usuarios en forma de variables, alias y funciones.

La serie exacta de archivos de inicio depende de un parámetro muy importante: El tipo de shell. Echemos un vistazo a la variedad de shell que existen.

Tipos de shell: Interactivo vs. No Interactivo y Inicio de sesión vs Sin inicio de sesión

Comencemos aclarando los conceptos de *interactivo* e *inicio de sesión* en el contexto de shells:

Shells interactivos / no interactivos

Este tipo de shell se refiere a la intercomunicación entre el usuario y el shell: Mediante el teclado el usuario proporciona la entrada digitando comandos en la terminal y su vez el shell proporciona la salida imprimiendo mensajes en la pantalla.

Shells de inicio de sesión / Sin inicio de sesión

Este tipo de shell se refiere al evento de un usuario cuando accede a un sistema informático por medio sus credenciales, como el nombre de usuario y la contraseña.

Tanto los shells interactivos como los no interactivos pueden ser de inicio de sesión o sin inicio de sesión y cualquier combinación posible de estos tipos tiene sus usos específicos.

shells Interactivo de inicio de sesión se ejecutan cuando los usuarios se conectan al sistema y se utilizan para personalizar las configuraciones de los usuarios según sus necesidades. Un buen ejemplo de este tipo de shell, sería el de un grupo de usuarios que pertenecen al mismo departamento y necesitan un conjunto de variables determinadas en sus sesiones.

Para *shells Interactivos sin inicio de sesión* nos referimos a cualquier otro shell abierto por el usuario después de entrar en el sistema. Los usuarios utilizan estos shells durante las sesiones para llevar a cabo tareas administrativas y de mantenimiento como la configuración de variables, el tiempo, la copia de archivos, crear scripts, etc.

Por otro lado, los shells no interactivos no requieren ningún tipo de interacción humana. Por lo tanto, estos shells no solicitan al usuario una entrada y su salida — si la hubiera — será escrita en un registro (en la mayoría de los casos).

Los *shells de inicio de sesión no interactivos* son bastante raros y poco prácticos. Sus usos son virtualmente inexistentes y sólo los comentaremos para comprender su comportamiento. Algunos ejemplos extraños incluyen forzar un script a ser ejecutado desde un shell de inicio de sesión con `/bin/bash --login <some_script>` o canalizar la salida estándar (*stdout*) de un comando a la entrada estándar (*stdin*) de una conexión ssh:

```
<some_command> | ssh <some_user>@<some_server>
```

En cuanto el shell *interactivo sin inicio de sesión* no hay interacción ni login en nombre del usuario, por lo que aquí nos referimos al uso de scripts automatizados. Estos scripts se utilizan

principalmente para llevar a cabo tareas administrativas y de mantenimiento repetitivas como las incluidas en los cronjobs. En algunos casos, `bash` no lee ningún archivo de inicio.

Iniciando una terminal

Cuando estamos en un entorno de escritorio, podemos abrir una terminal o cambiar a una de las consolas del sistema. Por lo tanto, un nuevo shell es un `pts` cuando se abre desde un emulador de terminal en el GUI o una `tty` cuando se ejecuta desde una consola de sistema. En el primer caso no se trata de una terminal sino de un emulador de terminales. Como parte de las sesiones gráficas, los emuladores de terminales como *gnome-terminal* o *konsole* son muy amplios en características y fáciles de usar en comparación con las terminales de interfaz de usuario basadas en texto. Los emuladores de terminal menos extenso en características incluyen — entre otros — *XTerm* y *sakura*.

Usando las teclas Ctrl + Alt + F1 - F6 podemos ir a los inicios de sesión de la consola que abren un shell de inicio de sesión interactivo basado en texto y la combinación de Ctrl + Alt + F7 llevará la sesión de vuelta al escritorio.

NOTE `tty` significa "teletypewritter"; `pts` significa "pseudo terminal slave". Para más información: `man tty` y `man pts`.

Ejecutando shells con `bash`

Después de iniciar sesión, escribe `bash` en una terminal para abrir un nuevo shell. Técnicamente, este shell es un proceso hijo del shell actual.

Al iniciar el proceso hijo de `bash`, podemos especificar varias opciones para definir qué tipo de shell queremos iniciar. Aquí hay algunas importantes a la hora invocarlo:

`bash -l` o **`bash --login`**
: Invocará un shell de inicio de sesión.

`bash -i`
: Invocará un shell interactivo.

`bash --noprofile`
: Con shell de inicio de sesión ignorará tanto el archivo de inicio de todo el sistema `/etc/profile` como los archivos de inicio a nivel de usuario `~/.bash_profile`, `~/.bash_login` y `~/.profile`.

`bash --norc`
: Con shell interactivo ignorará tanto el archivo de inicio del sistema `/etc/bash.bashrc` como el

archivo de inicio a nivel de usuario `~/.bashrc`.

`bash --rcfile <file>`

Con shell interactivo tomará `<file>` como el archivo de inicio ignorando a nivel de sistema `etc/bash.bashrc` y a nivel de usuario `~/.bashrc`.

Discutiremos los diferentes archivos de inicio a continuación.

Ejecutando shells con `su` y `sudo`

A través del uso de estos dos programas (similares) podemos obtener tipos específicos de shells:

`su`

Cambia el ID de usuario o lo convierte en superusuario (`root`). Con este comando podemos invocar ambos shells, el de inicio de sesión y sin inicio de sesión:

- `su - user2`, `su -l user2` o `su --login user2` iniciará un shell de inicio de sesión interactivo como `user2`.
- `su user2` iniciará un shell interactivo y sin inicio de sesión como `user2`.
- `su - root` o `su -` iniciará un shell de inicio de sesión interactivo como `root`.
- `su root` o `su` iniciará un shell interactivo y sin inicio de sesión como `root`.

`sudo`

Ejecuta comandos como otro usuario (incluyendo el superusuario). Debido a que este comando se usa principalmente para obtener privilegios temporales de root, el usuario que lo use debe estar en el archivo `sudoers`. Para añadir usuarios a `sudoers` necesitamos convertirnos en `root` y luego ejecutar:

```
root@debian:~# usermod -aG sudo user2
```

Así como `su`, `sudo` nos permite invocar tanto los shells de inicio de sesión como los de no de inicio de sesión:

- `sudo su - user2`, `sudo su -l user2` o `sudo su --login user2` iniciará un shell de inicio de sesión interactivo como `user2`.
- `sudo su user2` iniciará un shell interactivo sin inicio de sesión como `user2`.
- `sudo -u user2 -s` iniciará un shell interactivo sin inicio de sesión como `user2`.

- `sudo su - root` or `sudo su -` iniciará un shell de inicio de sesión interactivo como `root`.
- `sudo -i` iniciará un shell de inicio de sesión interactivo como `root`.
- `sudo -i <some_command>` iniciará un shell de inicio de sesión interactivo como `root`, ejecuta el comando y volverá al usuario original.
- `sudo su root` or `sudo su` iniciará un shell interactivo sin inicio de sesión como `root`.
- `sudo -s` or `sudo -u root -s` iniciará un shell sin inicio de sesión como`root`.

Cuando se usa `su` o `sudo`, es importante considerar el inicio de un nuevo shell y preguntarnos: ¿Necesitamos el entorno del usuario o no? Si es así, usaríamos las opciones que invocan las shells de inicio de sesión; si no, las que invocan sin inicio de sesión.

¿Qué tipo de shell tenemos?

Para saber en qué tipo de shell estamos trabajando, podemos escribir `echo $0` en la terminal y obtener la siguiente salida:

Inicio de sesión interactivo

`-bash` or `-su`

Sin inicio de sesión interactivo

`bash` or `/bin/bash`

Sin inicio de sesión no interactivo (scripts)

`<name_of_script>`

¿Cuántas shell tenemos?

Para observar cuántos `bash` shells tenemos en ejecutando en el sistema, podemos usar el comando `ps aux | grep bash`:

```
user2@debian:~$ ps aux | grep bash
user2       5270  0.1  0.1  25532  5664 pts/0    Ss   23:03   0:00 bash
user2       5411  0.3  0.1  25608  5268 tty1     S+   23:03   0:00 -bash
user2       5452  0.0  0.0  16760   940 pts/0    S+   23:04   0:00 grep
--color=auto bash
```

El `user2` en `debian` ha entrado en una sesión de GUI (Sistema de Ventanas X o en ingles: X Window System) y ha abierto *gnome-terminal*, luego ha pulsado `Ctrl`+`Alt`+`F1` para entrar en una sesión terminal `tty`. Finalmente, ha vuelto a la sesión del GUI presionando `Ctrl`+`Alt`+`F7` y ha escrito el

comando `ps aux | grep bash`. De esta manera, la salida muestra un shell interactivo sin inicio de sesión a través del emulador de terminal (`pts/0`) y un shell de inicio de sesión interactivo a través de la propia terminal basada en texto (`tty1`). Note también como el último campo de cada línea (el comando) es `bash` para el primero y `-bash` para el segundo.

¿De donde shell obtiene la configuración: Archivos de inicio?

Ahora que conocemos los tipos de shell que podemos encontrar en un sistema Linux, ya es hora de que veamos qué archivos de inicio son ejecutados por shell. Nóte que los scripts de todo el sistema o globales se colocan en el directorio `/etc/`, mientras que los locales o de nivel de usuario se encuentran en el directorio "home" del usuario (~). Además, cuando hay más de un archivo que buscar, y una vez que ellos es encontrado y ejecutado, los otros serán ignorados. Explora y estudia estos archivos con tu editor de texto favorito o escribiendo `less <fichero_de_inicio>`.

NOTE Los archivos de inicio se pueden dividir en Bash específicos (los que se limitan sólo a las configuraciones y comandos `bash`) y generales (relacionados con la mayoría de los shells).

Shell interactivo de inicio de sesión

Nivel Global

`/etc/profile`

Este es el archivo `.profile` de todo el sistema para el shell Bourne y los shells compatibles con Bourne (Incluido `bash`). A través de una serie de declaraciones `if` este archivo establece un número de variables como `PATH` y `PS1`, así como origen — si existe — tanto del archivo `/etc/bash.bashrc` como los del directorio `/etc/profile.d`.

`/etc/profile.d/*`

Este directorio puede contener scripts que son ejecutados por `/etc/profile`.

Nivel Local

`~/.bash_profile`

Este archivo específico de Bash se utiliza para configurar el entorno del usuario. También puede ser usado para crear el `~/.bash_login` y `~/.profile`.

`~/.bash_login`

Este archivo (específicamente), sólo se ejecutará si no hay un archivo `~/.bash_profile`. Su nombre sugiere que debería ser usado para ejecutar los comandos necesarios para el inicio de

sesión.

`~/.profile`

Este archivo no es específico de Bash y se obtiene sólo si no existe `~/.bash_profile` ni `~/.bash_login`, que es lo que normalmente ocurre. Por lo tanto, el propósito principal de `~/.profile` es el de revisar si se está ejecutando un shell de Bash, y si fuese afirmativo, obtener `~/.bashrc` (si existe). Normalmente establece la variable `PATH` para que incluya el directorio privado del usuario `~/bin` (si existe).

`~/.bash_logout`

Si existe, este archivo específico de Bash hace algunas operaciones de limpieza al salir del shell. Esto puede ser conveniente en casos como los de las sesiones remotas.

Exploración de los archivos de configuración de bash de inicio de sesión interactivo

Mostramos algunos de estos archivos en acción modificando `/etc/profile` y `/home/user2/.profile`. Añadiremos a cada uno una línea que nos recuerde el archivo que se está ejecutando:

```
root@debian:~# echo 'echo Hello from /etc/profile' >> /etc/profile
root@debian:~# echo 'echo Hello from ~/.profile' >> ~/.profile
```

NOTE Dos operadores de redirección `>>` añaden la salida de un comando a un archivo existente, sin sobrescribirlo. Sin embargo, si el archivo no existe, será creado.

Por lo tanto, a través de la salida de sus respectivos comandos `echo` sabremos cuando uno de estos archivos es leído y ejecutado. Para probarlo, veamos qué pasa cuando `user2` se conecta vía `ssh` desde otra máquina:

```
user2@debian:~$ ssh user2@192.168.1.6
user2@192.168.1.6's password:
Linux debian 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

Los programas incluidos en el sistema Debian GNU/Linux son software libre; los
términos exactos de distribución de cada programa se describen en archivos
individuales en /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 27 19:57:19 2018 from 192.168.1.10
Hello from /etc/profile
Hello from /home/user2/.profile
```

Como se observan en las dos últimas líneas, funcionó. Además, note tres cosas:

- El archivo global se ejecutó primero.
- No había archivos `.bash_profile` o `.bash_login` en el directorio "home" de `user2`.
- La tilde (~) se expandió a la ruta absoluta del archivo (`/home/user2/.profile`).

Shell Interactivo sin inicio de sesión

Nivel Global

`/etc/bash.bashrc`

Este es el archivo `.bashrc` de todo el sistema para los shells interactivos `bash`. A través de su ejecución, `bash` se asegura de que se está ejecutando interactivamente, comprueba el tamaño de la ventana después de cada comando (actualizando los valores de `LÍNEAS` y `COLUMNAS`, si es necesario) y establece algunas variables.

Nivel Local

`~/.bashrc`

Además de llevar a cabo tareas similares a las descritas para `/etc/bash.bashrc` a nivel de usuario (como comprobar el tamaño de la ventana o si se está ejecutando de forma interactiva), este archivo específico de Bash suele establecer algunas variables de historial y origen `~/.bash_aliases` (si existe). Aparte de eso, este archivo se utiliza normalmente para almacenar alias y funciones específicas de los usuarios.

Asimismo, también vale la pena señalar que `~/.bashrc` se lee si `bash` detecta que su `<stdin>` es

una conexión de red (como en ejemplo de la conexión *Secure Shell* (SSH)).

Exploración de los archivos de configuración de shell no Interactivo y de inicio de sesión

Modifiquemos ahora `/etc/bash.bashrc` y `/home/user2/.bashrc`:

```
root@debian:~# echo 'echo Hello from /etc/bash.bashrc' >> /etc/bash.bashrc
root@debian:~# echo 'echo Hello from ~/.bashrc' >> ~/.bashrc
```

Y esto es lo que sucede cuando `user2` comienza un nuevo shell:

```
user2@debian:~$ bash
Hello from /etc/bash.bashrc
Hello from /home/user2/.bashrc
```

De nuevo, los dos archivos fueron leídos y ejecutados.

WARNING Recuerde, debido al orden en que se ejecutan los archivos, los archivos locales tienen prioridad sobre los globales.

Shell no Interactivo de inicio de sesión

Un shell no interactivo con las opciones `-l` o `--login` es forzado a comportarse como un shell de inicio de sesión y así los archivos de inicio a ser ejecutados serán los mismos que los de los shells de inicio de sesión interactivos.

Para probarlo, escribamos un simple script y hagámoslo ejecutable. No incluiremos ningún *shebangs* porque invocaremos el ejecutable *bash* (`/bin/bash` con la opción de inicio de sesión) desde la línea de comandos.

1. Creamos el script `test.sh` que contiene la línea `echo 'Hello from a script'` para que podamos probar que el script se ejecuta con éxito:

   ```
   user2@debian:~$ echo "echo 'Hello from a script'" > test.sh
   ```

2. Hacemos que nuestro script sea ejecutable:

```
user2@debian:~$ chmod +x ./test.sh
```

3. Finalmente, invocamos a `bash` con la opción `l` para ejecutar el script:

```
user2@debian:~$ bash -l ./test.sh
Hello from /etc/profile
Hello from /home/user2/.profile
Hello from a script
```

¡Funciona! Antes de ejecutar el script, el login tuvo lugar y tanto el `/etc/profile` como el `~/.profile` fueron ejecutados.

NOTE Aprenderemos sobre *shebangs* y todos los demás aspectos del shell scripting en futuras lecciones.

Tengamos ahora la salida estándar (*stdout*) del comando `echo` en la entrada estándar (*stdin*) de una conexión `ssh` por medio de un pipe (`|`):

```
user2@debian:~$ echo "Hello-from-a-noninteractive-login-shell" | ssh
user2@192.168.1.6
Pseudo-terminal will not be allocated because stdin is not a terminal.
user2@192.168.1.6's password:
Linux debian 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

Los programas incluidos en el sistema Debian GNU/Linux son software libre; los
términos exactos de distribución de cada programa se describen en archivos
individuales en /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Hello from /etc/profile
Hello from /home/user2/.profile
-bash: line 1: Hello-from-a-noninteractive-login-shell: command not found
```

Una vez más, `/etc/profile` y `~/.profile` se ejecutan. Aparte de eso, la primera y la última línea de la salida son bastante reveladoras en lo que respecta al comportamiento de shell. ===== Shell no Interactivo sin inicio de sesión Los scripts no leen ninguno de los archivos listados arriba, pero buscan la variable de entorno `BASH_ENV` que expanden su valor si es necesario y la usan como el nombre de un archivo de inicio para leer y ejecutar comandos. Aprenderemos más sobre las *variables*

de entorno en la próxima lección.

Como se mencionó anteriormente, típicamente `/etc/profile` y `~/.profile` se aseguran de que tanto `/etc/bash.bashrc` como `~/.bashrc` se ejecuten después de un inicio de sesión exitoso. La salida del siguiente comando muestra este fenómeno:

```
root@debian:~# su - user2
Hello from /etc/bash.bashrc
Hello from /etc/profile
Hello from /home/user2/.bashrc
Hello from /home/user2/.profile
```

Teniendo en cuenta las líneas que hemos añadido previamente a los scripts de inicio e invocando un shell de inicio de sesión interactivo a nivel de usuario con `su - user2` las cuatro líneas de salida pueden explicarse de la siguiente manera:

1. `Hello from /etc/bash.bashrc` significa `/etc/profile` se ha obtenido `/etc/bash.bashrc`.
2. `Hello from /etc/profile` significa `/etc/profile` ha sido completamente leído y ejecutado.
3. `Hello from /home/user2/.bashrc` significa `~/.profile` se ha obtenido `~/.bashrc`.
4. `Hello from /home/user2/.profile` significa `~/.profile` ha sido completamente leído y ejecutado.

Note en como con `su - <nombre de usuario>` (también `su -l <nombre de usuario>` y `su --login <nombre de usuario>`) garantizamos la invocación de un shell de acceso, mientras que `su <nombre de usuario>` sólo habría invocado `/etc/bash.bashrc` y `~/.bashrc`.

Archivos fuentes

En las secciones anteriores hemos discutido que algunos scripts de inicio incluyen o ejecutan otros scripts. Este mecanismo se llama "sourcing" y se explica en esta sección.

Ejecutando archivos con `.`

El punto (`.`) se encuentra normalmente en los archivos de inicio.

En el archivo `.profile` de nuestro servidor de Debian podemos encontrar un ejemplo en el siguiente bloque:

```
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
    . "$HOME/.bashrc"
    fi
```

Hemos observado cómo la ejecución de un script puede llevar a la de otro. Así la declaración `if` garantiza que el archivo `$HOME/.bashrc` — si existe (`-f`) — se obtendrá (es decir, se leerá y se ejecutará) en el inicio de sesión:

```
. "$HOME/.bashrc"
```

NOTE Como aprenderemos en la próxima lección, `$HOME` es una variable de entorno que se expande a la ruta absoluta del directorio principal del usuario.

Además, podemos usar el `.` siempre que hayamos modificado un archivo de inicio y queramos hacer efectivos los cambios sin necesidad de reiniciar. Por ejemplo:

- Agregar un alias a `~/.bashrc`:

```
user2@debian:~$ echo "alias hi='echo We salute you.'" >> ~/.bashrc
```

WARNING Al enviar la salida de un comando a un archivo, recuerde no confundir el añadir (`>>`) con la sobreescritura (`>`).

- Imprime la última línea de `~/.bashrc` para comprobar que todo ha ido bien:

```
user2@debian:~$ tail -n 1 !$
tail -n 1 ~/.bashrc
alias hi='echo We salute you.'
```

NOTE `!$` se expande hasta el último argumento del comando anterior, en nuestro caso: `~/.bashrc`.

- ejecutar el archivo manuamente:

```
user2@debian:~$ . ~/.bashrc
```

- e invocar el alias para probar que funciona:

```
user2@debian:~$ hi
We salute you.
```

NOTE Refiérase a la siguiente lección para aprender acerca de *alias* y *variables*.

Ejecutando archivos con `source`

El comando "source" es un sinónimo de `.`. Así que para ejecutar `~/.bashrc` también podemos hacerlo de esta manera:

```
user2@debian:~$ source ~/.bashrc
```

El origen de los archivos de inicio de Shell: `SKEL`

`SKEL` es una variable cuyo valor es la ruta absoluta al directorio `skel`. Este directorio sirve como plantilla para la estructura del sistema de archivos de los principales directorios de los usuarios. Incluye los archivos que serán heredados por cualquier nueva cuenta de usuario que se cree (incluyendo, por supuesto, los archivos de configuración de los shells). El `SKEL` y otras variables relacionadas se almacenan en el `/etc/adduser.conf`, que es el archivo de configuración para `adduser`:

```
user2@debian:~$ grep SKEL /etc/adduser.conf
# La variable SKEL especifica el directorio que contiene el usuario "skeletal"...
SKEL=/etc/skel
# Si SKEL_IGNORE_REGEX está configurado, adduser ignorará los archivos que
coincidan con este.
SKEL_IGNORE_REGEX="dpkg-(old|new|dist|save)"
```

`SKEL` está configurado como `/etc/skel`; por lo tanto, los scripts de inicio que configuran nuestros shells están ahí:

```
user2@debian:~$ ls -a /etc/skel/
.  ..  .bash_logout  .bashrc  .profile
```

WARNING Recuerde, los archivos que empiezan por `.` están ocultos, por lo que debemos usar `ls -a` para verlos al listar el contenido del directorio.

Vamos a crear un directorio en `/etc/skel` para que todos los nuevos usuarios almacenen sus scripts personales:

1. Como `root` nos movemos a `/etc/skel`:

```
root@debian:~# cd /etc/skel/
root@debian:/etc/skel#
```

2. Listamos su contenido:

```
root@debian:/etc/skel# ls -a
.  ..  .bash_logout  .bashrc  .profile
```

3. Creamos nuestro directorio y comprobamos que todo ha ido como se esperaba:

```
root@debian:/etc/skel# mkdir my_personal_scripts
root@debian:/etc/skel# ls -a
.  ..  .bash_logout  .bashrc  my_personal_scripts  .profile
```

4. Ahora borramos `user2` junto con su directorio `home`:

```
root@debian:~# deluser --remove-home user2
Looking for files to backup/remove ...
Removing files ...
Removing user `user2' ...
Warning: group `user2' has no more members.
Done.
```

5. Añadimos `user2` de nuevo para que tenga un nuevo directorio principal:

```
root@debian:~# adduser user2
Adding user `user2' ...
Adding new group `user2' (1001) ...
Adding new user `user2' (1001) with group `user2' ...
Creating home directory `/home/user2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user2
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

6. Finalmente, entramos como `user2` y listamos todos los archivos en `/home/user2` para ver si todo salió como se esperaba:

```
root@debian:~# su - user2
user2@debian:~$ pwd
/home/user2
user2@debian:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  my_personal_scripts  .profile
```

Lo hizo.

Ejercicios guiados

1. Estudie cómo se han iniciado los shells en la columna "Iniciado con..." y complete la información requerida:

Iniciado con...	**Interactivo?**	**Inicio de sesión?**	**Resultado de `echo $0`**
`sudo ssh user2@machine2`			
`Ctrl` + `Alt` + `F2`			
`su - user2`			
gnome-terminal			
Un usuario regular usa *konsole* para iniciar una instancia de *sakura*			
Un script llamado `test.sh` que contiene el comando `echo $0`.			

2. Escriba los comandos `su` y `sudo` para lanzar el shell especificado:

Shell de inicio de sesión interactivo como `user2`

`su`:

`sudo`:

Shell de inicio de sesión interactivo como `root`

`su`:

`sudo`:

Shell interactivo sin inicio de sesión como `root`

`su`:

`sudo`:

Shell interactivo sin inicio de sesión como `user2`

`su`:

`sudo`:

3. ¿Qué archivo de inicio se lee cuando se inicia el shell bajo “Tipo de shell”?

Tipo de shell	`/etc/profile`	`/etc/bash.bashrc`	`~/.profile`	`~/.bashrc`
Shell de inicio de sesión interactivo como `user2`				
Shell de inicio de sesión interactivo como `root`				
Shell interactivo sin inicio de sesión como `root`				
Shell interactivo sin inicio de sesión como `user2`				

Ejercicios de exploración

1. En Bash podemos escribir una simple función "¡Hola mundo!" incluyendo el siguiente código en un archivo vacío:

```
function hello() {
     echo "Hello world!"
}
```

   - ¿Qué deberíamos hacer a continuación para que la función esté disponible para shell?

   - Una vez que esté disponible para el shell actual, ¿cómo lo invocarías?

   - Para automatizar las cosas, ¿en qué archivo agregaría la función y su invocación para que se ejecute cuando `user2` abra una terminal de una sesión de Ventanas X (X Windows)? ¿Qué tipo de shell es?

   - ¿En qué archivo pondrías la función y su invocación para que se ejecute cuando `root` lance un nuevo shell interactivo independientemente de si es de inicio de sesión o no?

2. Observemos el siguiente script, `¡Hola mundo!` de `Bash`:

```
#!/bin/bash

#hello_world: a simple bash script to discuss interaction in scripts.

echo "Hello world!"
```

   - Supongamos que establecemos permisos de ejecución y lo ejecutamos. ¿Sería un script interactivo? ¿Por qué?

     ¿Qué hace que un script sea interactivo?

3. Imagina que has cambiado los valores de algunas variables en `~/.bashrc` y quieres que esos cambios surtan efecto sin reiniciar. Desde tu directorio principal, ¿cómo podrías lograrlo de dos maneras diferentes?

4. John acaba de iniciar una sesión de X wIndows en un servidor Linux. Abre un emulador de terminal para realizar algunas tareas administrativas pero, sorprendentemente, la sesión se congela y necesita abrir un shell de texto.
   - ¿Cómo puede abrir esa `tty`?
   - ¿Qué archivos de inicio se obtendrán?

5. Linda es una usuaria de un servidor Linux. Le pide amablemente al administrador que tenga un archivo `~/.bash_login` para que pueda tener la hora y la fecha impresa en la pantalla cuando se conecte. A otros usuarios les gusta la idea y siguen el ejemplo. El administrador tiene dificultades para crear el archivo para los otros usuarios del servidor, así que decide añadir una nueva política y crear un archivo `~/.bash_login` para todos los nuevos usuarios. ¿Cómo puede el administrador realizar esa tarea?

Resumen

En esta lección aprendimos:

- Los Shells establecen el entorno de los usuarios en un sistema Linux.
- *Bash* es el shell número uno en todas las distribuciones de GNU/Linux.
- El primer trabajo que realiza un shell es leer y ejecutar uno o varios archivos de inicio.
- Los conceptos de *interacción* y *inicio de sesión* en relación con los shells.
- Cómo lanzar diferentes tipos de shells con `bash`, `su`, `sudo` y *Ctrl* + *Alt* + *F1*-*F6*.
- Cómo comprobar el tipo de shell con `echo $0`.
- Los archivos de inicio locales `~/.bash_profile`, `~/.profile`, `~/.bash_login`, `~/.bash_logout` y `~/.bashrc`.
- Los archivos de inicio globales `/etc/profile`, `/etc/profile.d/*`, `/etc/bash.bashrc`.
- Los archivos locales tienen prioridad sobre los globales.
- Cómo redirigir la salida de un comando con `>` (sobrescribir) y `>>` (añadir).
- El significado del directorio `skel`.
- Comprender el funcionamiento de source.

Comandos usados en esta lección:

`bash`
: Crear un nuevo shell.

`su`
: Crear un nuevo shell.

`sudo`
: Crear un nuevo shell.

`usermod`
: Modificar una cuenta de usuario.

`echo`
: Imprimir en pantalla una línea de texto.

`ps`
: Captura de los procesos actuales.

`less`
: Un visualizador para archivos largos.

`ssh`
: Inicie una conexión SSH (remotamente).

`chmod`
: Cambiar permisos de un archivo, por ejemplo hacerlo ejecutable.

`grep`
: Imprime líneas que coincidan con un patrón.

`ls`
: Lista el contenido de un directorio.

`cd`
: Cambiar de directorio.

`mkdir`
: Crear un directorio.

`deluser`
: Eliminar un usuario

`adduser`
: Crear un nuevo usuario.

`.`
: Ejecutar un archivo

`source`
: Realiza una ejecución de un archivo con el comando source

`tail`
: Emitir la última parte de los archivos.

Respuestas a los ejercicios guiados

1. Estudie cómo se han iniciado los shells en la columna "Iniciado con…" y complete la información requerida:

Iniciado con…	Interactivo?	Inicio de sesión?	Resultado de `echo $0`
`sudo ssh user2@machine2`	Si	Si	`-bash`
`Ctrl` + `Alt` + `F2`	Si	Si	`-bash`
`su - user2`	Si	Si	`-bash`
gnome-terminal	Si	No	`bash`
Un usuario regular usa *konsole* para iniciar una instancia de *sakura*	Si	No	`/bin/bash`
Un script llamado `test.sh` contiene el comando `echo $0`	No	No	`./test.sh`

2. Escriba los comandos `su` y `sudo` para lanzar el shell especificado:

Shell de inicio de sesión interactivo como `user2`

`su`

`su - user2`, `su -l user2` o `su --login user2`

`sudo`

`sudo su - user2`, `sudo su -l user2` o `sudo su --login user2`

Shell de inicio de sesión interactivo como `root`

`su`

`su - root` o `su -`

`sudo`

`sudo su - root`, `sudo su -` o `sudo -i`

Shell interactivo de no inicio de sesion como `root`

`su`

`su root` o `su`

`sudo`

`sudo su root`, `sudo su`, `sudo -s` o `sudo -u root -s`

Shell interactivo de no inicio de sesion como `user2`

`su`

`su user2`

`sudo`

`sudo su user2` o `sudo -u user2 -s`

3. ¿Qué archivo de inicio se lee cuando se inicia el shell bajo “Tipo de shell”?

Tipo de shell	`/etc/profile`	`/etc/bash.bashrc`	`~/.profile`	`~/.bashrc`
Shell de inicio de sesión interactivo como `user2`	Si	Si	Si	Si
Shell de inicio de sesión interactivo como `root`	Si	Si	No	No
Shell interactivo de no inicio de sesion como `root`	No	Si	No	No
Shell interactivo de no inicio de sesion como `user2`	No	Si	No	Yes

Respuestas a los ejercicios de exploración

1. In Bash we can write a simple `Hello world!` function by including the following code in an empty file:

```
function hello() {
     echo "Hello world!"
}
```

   - What should we do next to make the function available to the shell?

     To make the function availabe to the current shell, we must source the file which includes it.

   - Once it is available to the current shell, how would you invoke it?

     We will invoke it by typing its name into the terminal.

   - To automate things, in what file would you put the function and its invocation so that it gets executed when `user2` opens a terminal from an X Window session? What type of shell is it?

     The best file to put it is `/home/user2/.bashrc`. The invoked shell would be an interactive non-login one.

   - In what file would you put the function and its invocation so that it is run when `root` launches a new interactive shell irrespective of whether it is login or not?

     In `/etc/bash.bashrc` since this file gets executed for all interactive shells — whether login or not.

2. Observemos el siguiente script, `¡Hola mundo!` de `Bash`:

```
#!/bin/bash

#hello_world: a simple bash script to discuss interaction in scripts.

echo "Hello world!"
```

   - Supongamos que establecemos permisos de ejecución y lo ejecutamos. ¿Sería un script interactivo? ¿Por qué?

     No, ya que no hay interacción humana y no hay comandos que sean tecleados por el

usuario.

¿Qué hace que un script sea interactivo?

El hecho de que requiere la entrada del usuario.

3. Imagina que has cambiado los valores de algunas variables en `~/.bashrc` y quieres que esos cambios surtan efecto sin reiniciar. Desde tu directorio principal, ¿cómo podrías lograrlo de dos maneras diferentes?

```
$ source .bashrc
```

or

```
$ . .bashrc
```

4. John acaba de iniciar una sesión de X window en un servidor Linux. Abre un emulador de terminal para realizar algunas tareas administrativas pero, sorprendentemente, la sesión se congela y necesita abrir un shell de texto.
   - ¿Cómo puede abrir esa `tty`?

     Podría hacerlo presionando `Ctrl` + `Alt` + `F1`-`F6` para entrar en una de las seis tty.

   - ¿Qué archivos de inicio se obtendrán?

     `/etc/profile`
     `/home/john/.profile`

5. Linda es una usuaria de un servidor Linux. Le pide amablemente al administrador que tenga un archivo `~/.bash_login` para que pueda tener la hora y la fecha impresa en la pantalla cuando se conecte. A otros usuarios les gusta la idea y siguen el ejemplo. El administrador tiene dificultades para crear el archivo para los otros usuarios del servidor, así que decide añadir una nueva política y crear un archivo `~/.bash_login` para todos los nuevos usuarios. ¿Cómo puede el administrador realizar esa tarea?

   Podría lograrlo poniendo `.bash_login` en el directorio `/etc/skel`.



105.1 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	105 Shells y scripts
Objetivo:	105.1 Personalizar y usar el entorno de shell
Lección:	2 de 3

Introducción

Piense en una variable como una caja imaginaria en la que coloca temporalmente una información. Al igual que los scripts de inicialización, Bash clasifica las variables como *shell/local* (las que se ubican sólo dentro de los límites del shell en el que fueron creadas) o *entorno/global* (las que son heredadas por shells y/o procesos hijos). De hecho, en la lección anterior hemos echado un vistazo a las shell y sus scripts de configuración o inicialización. Ahora es conveniente señalar el poder de estos archivos de inicio, el cual radica en el hecho de que nos permiten utilizar variables — así como alias y funciones — que nos ayudan a crear y personalizar el entorno de shell como queramos.

Variables: Asignación y referencia

Una variable puede definirse como un nombre que contiene un valor.

En Bash, dar un valor a un nombre se llama *asignación de variables* y es la forma en que creamos o establecemos las variables. Por otro lado, el proceso de acceder al valor contenido en el nombre se llama *variable referenciada.*

La sintaxis para la asignación de variables es:

```
<variable_name>=<variable_value>
```

Por ejemplo:

```
$ distro=zorinos
```

La variable `distro` es igual a `zorinos`, es decir, hay una porción de memoria que contiene el valor `zorinos` — con `distro` siendo el puntero hacia este.

Tenga en cuenta, que no puede haber ningún espacio a ambos lados del signo igual cuando se asigna una variable:

```
$ distro =zorinos
-bash: distro: command not found
$ distro= zorinos
-bash: zorinos: command not found
```

A causa de nuestro error, Bash leyó `distro` y `zorinos` como órdenes.

Para referenciar una variable (es decir, para comprobar su valor) utilizamos el comando `echo` que precede al nombre de la variable con un signo `$`:

```
$ echo $distro
zorinos
```

Nombres de variables

Al elegir el nombre de las variables, hay ciertas reglas que debemos tener en cuenta.

El nombre de una variable puede contener letras (`a-z`,`A-Z`), números (`0-9`) y guiones bajos (`_`):

```
$ distro=zorinos
$ echo $distro
zorinos
$ DISTRO=zorinos
$ echo $DISTRO
zorinos
$ distro_1=zorinos
$ echo $distro_1
zorinos
$ _distro=zorinos
$ echo $_distro
zorinos
```

No puede empezar con un numero o Bash se confundira:

```
$ 1distro=zorinos
-bash: 1distro=zorinos: command not found
```

No puede contener espacios (ni siquiera usando comillas); por convención, se usan los guiones bajos en su lugar:

```
$ "my distro"=zorinos
-bash: my: command not found
$ my_distro=zorinos
$ echo $my_distro
zorinos
```

Valores de las variables

En lo que respecta a la referencia o el valor de las variables, también es importante considerar una serie de reglas.

Las variables pueden contener cualquier carácter alfanumérico (`a-z`,`A-Z`,`0-9`) así como la mayoría de los caracteres (`?`,`!`,`*`,`.`,`/`, etc.):

```
$ distro=zorin12.4?
$ echo $distro
zorin12.4?
```

Los valores de las variables deben ser encerrados entre comillas si contienen espacios simples:

```
$ distro=zorin 12.4
-bash: 12.4: command not found
$ distro="zorin 12.4"
$ echo $distro
zorin 12.4
$ distro='zorin 12.4'
$ echo $distro
zorin 12.4
```

Los valores de las variables también deben ser encerrados entre comillas si contienen caracteres como los utilizados para la redirección (`<`,`>`) o el símbolo de "pipe" (`|`). Lo único que hace el siguiente comando es crear un archivo vacío llamado `zorin`:

```
$ distro=>zorin
$ echo $distro

$ ls zorin
zorin
```

Esto funciona, siempre y cuando usemos las comillas:

```
$ distro=">zorin"
$ echo $distro
>zorin
```

Sin embargo, las comillas simples y dobles no siempre son intercambiables. Según lo que hagamos con una variable (asignación o referencia), el uso de una u otra tiene implicaciones y dará resultados diferentes. En el contexto de la asignación de variables, las comillas simples toman todos los caracteres del valor de la variable *literalmente*, mientras que las comillas dobles permiten la sustitución de la variable:

```
$ lizard=uromastyx
$ animal='My $lizard'
$ echo $animal
My $lizard
$ animal="My $lizard"
$ echo $animal
My uromastyx
```

Por otra parte, cuando se hace referencia a una variable cuyo valor incluye algunos espacios iniciales (o adicionales) — a veces combinados con asteriscos — es obligatorio utilizar comillas dobles después del comando `echo` para evitar la *división de campos* y la *expansión de nombres*:

```
$ lizard="   genus   |   uromastyx"
$ echo $lizard
genus | uromastyx
$ echo "$lizard"
   genus   |   uromastyx
```

Si la referencia de la variable contiene un signo de exclamación de cierre, éste debe ser el último carácter de la cadena (ya que de lo contrario Bash pensará que nos referimos a un evento `histórico`):

```
$ distro=zorin.?/!os
-bash: !os: event not found
$ distro=zorin.?/!
$ echo $distro
zorin.?/!
```

Cualquier "backslash" debe escapar por otro. Además, si una barra invertida es el último caracter de la cadena (string) y no escapa, Bash interpretará que queremos un salto de línea y nos dará una nueva línea:

```
$ distro=zorinos\
>
$ distro=zorinos\\
$ echo $distro
zorinos\
```

En las próximas dos secciones resumiremos las principales diferencias entre las variables *locales* y de *entorno*.

Variables locales o de Shell

Las variables locales o de shell existen sólo en el shell en el que se crean. Por convención, las variables locales se escriben en minúsculas.

Con el fin de realizar algunas pruebas, vamos a crear una variable local. Como se ha explicado anteriormente, elegimos un nombre apropiado para la variable y la equiparamos a un valor apropiado. Por ejemplo:

```
$ reptile=tortoise
```

Usemos ahora el comando `echo` para referirnos a nuestra variable y comprobar que todo ha ido como se esperaba:

```
$ echo $reptile
tortoise
```

En ciertos escenarios — cuando se escriben los scripts — la inmutabilidad puede ser una característica interesante de las variables. Si queremos que nuestras variables sean inmutables, podemos crearlas en modo `sólo lectura` (readonly):

```
$ readonly reptile=tortoise
```

Otra opción es convertirlas después de haberlas creado:

```
$ reptile=tortoise
$ readonly reptile
```

Ahora, si intentamos cambiar el valor de `reptile`, Bash lo negará:

```
$ reptile=lizard
-bash: distro: readonly variable
```

NOTE Para listar todas las variables de sólo lectura en nuestra sesión actual, escriba `readonly` o `readonly -p` en la terminal.

Un comando útil cuando se trata de variables locales es `set`.

`set` da salida a todas las variables y funciones de shell que se encuentran actualmente asignadas. Dado que pueden ser muchas líneas (¡pruébalo tú mismo!), se recomienda usarlo en combinación con un buscador como `less`:

```
$ set | less
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:fo
rce_fignore:histappend:interactive_comments:login_shell:progcomp:promptvars:sourcep
ath
BASH_ALIASES=()
BASH_ARGC=()
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_COMPAT_DIR=/etc/bash_completion.d
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="4" [1]="4" [2]="12" [3]="1" [4]="release" [5]="x86_64-pc-linux-
gnu")
BASH_VERSION='4.4.12(1)-release'
(...)
```

¿Se encuentra nuestra variable `reptile`?

```
$ set | grep reptile
reptile=tortoise
```

¡Sí, ahí está!

Sin embargo, `reptile`—siendo una variable local—no será heredado por ningún proceso hijo generado desde el shell actual:

```
$ bash
$ set | grep reptile
$
```

Y por supuesto, tampoco podemos ejecutar el comando `echo` para revisar su valor:

```
$ echo $reptile
$
```

NOTE Al teclear el comando `bash` en la terminal abrimos un nuevo shell (hijo).

Para remover cualquier variable (ya sea local o global), usamos el comando `unset`:

```
$ echo $reptile
tortoise
$ unset reptile
$ echo $reptile
$
```

NOTE `unset` debe ser seguido por el nombre de la variable solamente (no precedido por el símbolo $).

Variables globales o de entorno

Existen variables globales o de entorno para el shell actual, así como para todos los procesos subsecuentes que se derivan de este. Por convención, las variables de entorno se escriben en mayúsculas:

```
$ echo $SHELL
/bin/bash
```

Podemos pasar recursivamente el valor de estas variables a otras variables y el valor de estas últimas se expandirá finalmente a las primeras:

```
$ my_shell=$SHELL
$ echo $my_shell
/bin/bash
$ your_shell=$my_shell
$ echo $your_shell
/bin/bash
$ our_shell=$your_shell
$ echo $our_shell
/bin/bash
```

Para que una variable de shell local se convierta en una variable de entorno, se debe utilizar el comando `export`:

```
$ export reptile
```

Con `export reptile` hemos convertido nuestra variable local en una variable de entorno para que los shells hijos puedan reconocerla y usarla:

```
$ bash
$ echo $reptile
tortoise
```

De la misma manera, `export` puede ser usado para asignar y exportar una variable, todo a la vez:

```
$ export amphibian=frog
```

Ahora podemos abrir una nueva instancia de Bash y referirnos con éxito a la nueva variable:

```
$ bash
$ echo $amphibian
frog
```

NOTE Con `export -n <VARIABLE-NAME>` la variable se convertirá de nuevo en una variable de la shell local.

El comando `export` también nos dará una lista de todas las variables de entorno existentes o cuando se digita con la opción `-p`:

```
$ export
declare -x HOME="/home/user2"
declare -x LANG="en_GB.UTF-8"
declare -x LOGNAME="user2"
(...)
declare -x PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
declare -x PWD="/home/user2"
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_CLIENT="192.168.1.10 49330 22"
declare -x SSH_CONNECTION="192.168.1.10 49330 192.168.1.7 22"
declare -x SSH_TTY="/dev/pts/0"
declare -x TERM="xterm-256color"
declare -x USER="user2"
declare -x XDG_RUNTIME_DIR="/run/user/1001"
declare -x XDG_SESSION_ID="8"
declare -x reptile="tortoise"
```

NOTE El comando `declare -x` es equivalente a `export`.

Dos comandos más que pueden ser usados para imprimir una lista de todas las variables de entorno son `env` y `printenv`:

```
$ env
SSH_CONNECTION=192.168.1.10 48678 192.168.1.7 22
LANG=en_GB.UTF-8
XDG_SESSION_ID=3
USER=user2
PWD=/home/user2
HOME=/home/user2
SSH_CLIENT=192.168.1.10 48678 22
SSH_TTY=/dev/pts/0
MAIL=/var/mail/user2
TERM=xterm-256color
SHELL=/bin/bash
SHLVL=1
LOGNAME=user2
XDG_RUNTIME_DIR=/run/user/1001
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
_=/usr/bin/env
```

Además de ser un sinónimo de `env`, a veces podemos usar `printenv` de forma similar a como usamos

el comando `echo` para comprobar el valor de una variable:

```
$ echo $PWD
/home/user2
$ printenv PWD
/home/user2
```

Sin embargo, con `printenv` el nombre de la variable no está precedido por `$`.

NOTE `PWD` almacena la ruta del directorio de trabajo actual. Aprenderemos sobre esta y otras variables de entorno más adelante.

Ejecución de un programa en un entorno modificado

El comando `env` puede ser usado para modificar el entorno del shell en el momento de la ejecución de un programa.

Para iniciar una nueva sesión de Bash con un entorno tan vacío como sea posible — despejando la mayoría de las variables (así como las funciones y alias) — usaremos `env` con la opción `-i`:

```
$ env -i bash
```

Ahora la mayoría de las variables de nuestro entorno han desaparecido:

```
$ echo $USER
$
```

Y sólo quedan unas pocas:

```
$ env
LS_COLORS=
PWD=/home/user2
SHLVL=1
_=/usr/bin/printenv
```

También podemos usar `env` para establecer una variable particular para un programa particular.

En nuestra lección anterior, cuando hablamos de los shells no interactivos sin inicio de sesión,

observamos como los scripts no leen ningún archivo de inicio estándar sino que buscan el valor de la variable `BASH_ENV` y lo usan como su archivo de inicio si existe.

Demostrémos este proceso:

1. Creamos nuestro propio archivo de inicio llamado `.startup_script` con el siguiente contenido:

```
CROCODILIAN=caiman
```

2. Escribimos un script Bash llamado `test_env.sh` con el siguiente contenido:

```
#!/bin/bash

echo $CROCODILIAN
```

3. Establecemos el bit ejecutable para nuestro script `test_env.sh`:

```
$ chmod +x test_env.sh
```

4. Por último, usamos `env` para establecer `BASH_ENV` en `startup_script` para `test_env.sh`:

```
$ env BASH_ENV=/home/user2/.startup_script ./test_env.sh
caiman
```

El comando `env` está implícito incluso si nos deshacemos de este:

```
$ BASH_ENV=/home/user2/.startup_script ./test_env.sh
caiman
```

NOTE Si no comprendes la línea `#!/bin/bash` o el comando `chmod +x`, ¡no te asustes! Aprenderemos todo lo necesario acerca de los scripts de shell en futuras lecciones. Por ahora, sólo recuerda que para ejecutar un script desde su propio directorio usamos `./some_script`.

Variables comunes de entorno

Es hora de revisar algunas de las variables de entorno más relevantes que se establecen en los archivos de configuración de Bash.

`DISPLAY`

En relación con el servidor X, el valor de esta variable se compone normalmente de tres elementos:

- El hostname (la ausencia de este significa `localhost`) donde se ejecuta el servidor X.
- Dos puntos como delimitador.
- Un número (normalmente es `0` y se refiere a la pantalla de la computadora).

```
$ printenv DISPLAY
:0
```

Un valor vacío para esta variable significa un servidor sin un sistema X WIndow. Un número extra — como en `mi.xserver:0:1` — se referiría al número de pantalla (si existe más de uno).

`HISTCONTROL`

Esta variable controla qué comandos se guardan en `HISTFILE` (ver abajo). Hay tres valores posibles:

`ignorespace`

Los comandos que empiecen con un espacio no se guardarán.

`ignoredups`

Un comando que es el mismo que el anterior no se guardará.

`ignoreboth`

Los comandos que caen en cualquiera de las dos categorías anteriores no se guardarán.

```
$ echo $HISTCONTROL
ignoreboth
```

`HISTSIZE`

Esto establece el número de comandos que se almacenarán en la memoria mientras dure la sesión de shell.

```
$ echo $HISTSIZE
1000
```

HISTFILESIZE

Esto establece el número de comandos que se guardarán en `HISTFILE` tanto al principio como al final de la sesión:

```
$ echo $HISTFILESIZE
2000
```

HISTFILE

El nombre del archivo que almacena todos los comandos a medida que se escriben. Por defecto este archivo se encuentra en `~/.bash_history`:

```
$ echo $HISTFILE
/home/user2/.bash_history
```

NOTE Para ver el contenido de `HISTFILE`, simplemente escribimos `history`. Alternativamente, podemos especificar el número de comandos que queremos ver pasando un argumento (el número de los comandos más recientes) a `history`, ejemplo: `history 3`.

HOME

Esta variable almacena la ruta absoluta del directorio principal del usuario y se establece cuando el usuario se conecta.

Esta parte del código — de `~/.profile` — se explica por sí mismo (tiene como origen `"$HOME/.bashrc"` si existe):

```
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
    . "$HOME/.bashrc"
    fi
```

NOTE Si no comprende bien la afirmación `if`, no te preocupes: sólo consulta las lecciones sobre shell scripting.

Recuerda que `~` es equivalente a `$HOME`:

```
$ echo ~; echo $HOME
/home/carol
/home/carol
```

NOTE Los comandos pueden ser concatenados con un punto y coma (`;`).

También podemos probar esto con una declaración de `if`.

```
$ if [ ~ == "$HOME" ]; then echo "true"; else echo "false"; fi
true
```

NOTE Recuerde: El signo de igualdad `=` se utiliza para la asignación de variables. Y el signo `==` se usa para probar la igualdad.

`HOSTNAME`

Variable que almacena nombre del computador en la red:

```
$ echo $HOSTNAME
debian
```

`HOSTTYPE`

Esto almacena la arquitectura de la unidad central de procesamiento (CPU) del computador:

```
$ echo $HOSTTYPE
x86_64
```

`LANG`

Esta variable guarda la información de localización que utiliza el sistema:

```
$ echo $LANG
en_UK.UTF-8
```

`LD_LIBRARY_PATH`

Esta variable consiste en un conjunto de directorios separados por dos puntos donde las bibliotecas compartidas (shared libraries) son compartidas por los programas:

```
$ echo $LD_LIBRARY_PATH
/usr/local/lib
```

`MAIL`

Esta variable almacena el archivo en el que Bash revisa el correo electrónico:

```
$ echo $MAIL
/var/mail/carol
```

Otro valor común para esta variable es `/var/spool/mail/$USER`.

`MAILCHECK`

Esta variable almacena un valor numérico que indica en segundos la frecuencia con la que Bash comprueba si hay correo nuevo:

```
$ echo $MAILCHECK
60
```

`PATH`

Esta variable de entorno almacena la lista de directorios donde Bash busca los archivos ejecutables cuando se le indica que ejecute cualquier programa. En nuestra máquina de ejemplo, esta variable se establece a través del archivo `/etc/profile` de todo el sistema:

```
if [ "`id -u`" -eq 0 ]; then
  PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
  PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
export PATH
```

A través de la declaración `if` se comprueba la identidad del usuario y — dependiendo del resultado de la prueba (`root` o de otra manera) — obtendremos un `PATH` u otro. Finalmente, el `PATH` elegido se propaga invocando el comando `export`.

Observe dos cosas con respecto al valor de `PATH`:

- Los nombres de los directorios se escriben usando rutas absolutas.
- Los dos puntos se usan como delimitador.

  Si quisiéramos incluir la carpeta `/usr/local/sbin` en el `PATH` para usuarios habituales, modificaremos la línea para que se vea así:

```
(...)
else

PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/usr/local/sb
in"
fi
export PATH
```

  Ahora podemos ver cómo cambia el valor de la variable cuando entramos como usuario regular:

```
# su - carol
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/usr/local/sbin
```

NOTE También podríamos haber añadido `/usr/local/sbin` al `PATH` del usuario en la línea de comandos o bien `PATH=/usr/local/sbin:$PATH` o `PATH=$PATH:/usr/local/sbin` — el primero hace que `/usr/local/sbin` sea el primer directorio en el que se busquen archivos ejecutables; el segundo hace que sea el último.

PS1

Esta variable almacena el valor del indicador Bash. En la siguiente parte de código (también de `/etc/profile`), la sentencia `if` comprueba la identidad del usuario y en consecuencia brinda un indicador muy discreto (`#` para `root` o `$` para usuarios regulares):

```
if [ "`id -u`" -eq 0 ]; then
  PS1='# '
else
  PS1='$ '
fi
```

NOTE El `id` de `root` es `0`. Conviértete en `root` y compruébalo tú mismo con `id -u`.

Otras variables de aviso incluyen:

`PS2`
: Normalmente se establece en `>` y se usa como un mensaje de continuidad para comandos largos de varias líneas.

`PS3`
: Usado como el indicador para el comando `select`.

`PS4`
: Normalmente se establece en `+` y se usa para la depuración.

`SHELL`
: Esta variable almacena la ruta absoluta del shell actual:

```
$ echo $SHELL
/bin/bash
```

`USER`
: Esto almacena el nombre del usuario actual:

```
$ echo $USER
carol
```

Ejercicios guiados

1. Observe la asignación de la variable en la columna “Comando(s)” e indique si la variable resultante es “Local” o “Global”:

Comando(s)	Local	Global
`debian=mother`		
`ubuntu=deb-based`		
`mint=ubuntu-based; export mint`		
`export suse=rpm-based`		
`zorin=ubuntu-based`		

2. Estudie el “Comando” y la “Salida” y explica el significado:

Comando	Salida	Signficado
`echo $HISTCONTROL`	`ignoreboth`	
`echo ~`	`/home/carol`	
`echo $DISPLAY`	`reptilium:0:2`	
`echo $MAILCHECK`	`60`	
`echo $HISTFILE`	`/home/carol/.bash_history`	

3. Las variables están siendo puestas incorrectamente en la columna “Comando erróneo”. Proporcione la información que falta en “Comando correcto” y “Referencia de la variable” para que obtengamos la “Salida esperada”:

Comando erróneo	Comando correcto	Referencia de la variable	Salida esperada
`lizard =chameleon`			`chameleon`
`cool lizard=chameleon`			`chameleon`
`lizard=cha|me|leon`			`cha|me|leon`
`lizard=/** chameleon **/`			`/** chamelon **/`
`win_path=C:\path\to\dir\`			`C:\path\to\dir\`

4. Considere el propósito y escriba el comando apropiado:

Propósito	Comando
Establecer el lenguaje del actual shell al español UTF-8 (`es_ES.UTF-8`).	
Imprime el nombre del directorio actual.	
Referencia a la variable de entorno que almacena la información sobre las conexiones `ssh`.	
Establecer el `PATH` para incluir `/home/carol/scripts` como el último directorio para buscar ejecutables.	
Establecer el valor de`my_path` en `PATH`.	
Establecer el valor de `my_path` en el de `PATH`.	

5. Crear una variable local llamada "mammal" y asígnale el valor `gnu`:

6. Usando la sustitución de variables, cree otra variable local llamada `var_sub` con el valor apropiado para que cuando se haga referencia a través de `echo $var_sub` obtengamos: `The value of mammal is gnu`:

7. Convierte a `mammal` en una variable de entorno:

8. Búscalo con `set` y `grep`:

9. Búscalo con `env` y `grep`:

10. Crear, en dos comandos consecutivos, una variable de entorno llamada `BIRD` cuyo valor es `penguin`:

11. Crear en dos comandos consecutivos, una variable de entorno llamada `NEW_BIRD` cuyo valor es `yellow-eyed penguin`:

12. Asumiendo que eres `user2`, cree una carpeta llamada `bin` en tu directorio principal:

13. Escriba el comando para agregar la carpeta `~/bin` a su `PATH` para que sea la primera carpeta en la que `bash` busque binarios:

14. Para garantizar que el valor de `PATH` permanezca inalterado en los reinicios, ¿Qué parte del código — en forma de una declaración `if` — agregarías en el `~/.profile`?

Ejercicios de exploración

1. `let`: más que la evaluación de la expresión aritmética::
   - Haz una búsqueda en la página web de `let` y sus implicaciones al establecer las variables y crea una nueva variable local llamada `my_val` cuyo valor es `10` — como resultado de sumar 5 + 5:

   - Ahora crea otra variable llamada `your_val`, cuyo valor es `5` — como resultado de dividir el valor de `my_val` entre 2:

2. ¿El resultado de un comando en una variable? Por supuesto, eso es posible; se llama *sustitución de comandos*. Investiga y estudia la siguiente función llamada `music_info`:

```
music_info(){
latest_music=`ls -l1t ~/Music | head -n 6`
echo -e "Your latest 5 music files:\n$latest_music"
}
```

   El resultado del comando `ls -l1t ~/Music | head -n 6` se convierte en el valor de la variable `latest_music`. Luego, se hace referencia a la variable `latest_music` en el comando `echo` (que genera el número total de bytes ocupados por la carpeta `Music` y los últimos cinco archivos de música almacenados en la carpeta `Music` — uno por línea).

   ¿Cuál de los siguientes es un sinónimo válido?

```
latest_music=`ls -l1t ~/Music | head -n 6`
```

   Opción A:

```
latest_music=$(ls -l1t ~/Music| head -n 6)
```

   Opción B:

```
latest_music="(ls -l1t ~/Music| head -n 6)"
```

Opción C:

```
latest_music=((ls -l1t ~/Music| head -n 6))
```

Resumen

En esta lección aprendimos:

- Las variables son una parte muy importante del entorno del shell ya que son utilizadas por el propio shell así como por otros programas.
- Como asignar y referenciar las variables.
- Las diferencias entre las variables *local* y *global* (o *entorno*).
- Como hacer variables *sólo lectura*.
- Como convertir una variable local en una variable de entorno con el comando `export`.
- Como listar todas las variables de entorno.
- Como ejecutar un programa en un entorno modificado.
- Como hacer que las variables sean persistentes con la ayuda de los scripts de inicio.
- Algunas variables de entorno comunes: `DISPLAY`, `HISTCONTROL`, `HISTSIZE`, `HISTFILESIZE`, `HISTFILE`, `HOME`, `HOSTNAME`, `HOSTTYPE`, `LANG`, `LD_LIBRARY_PATH`, `MAIL`, `MAILCHECK`, `PATH`, `PS1` (y otras variables), `SHELL` y `USER`.
- El significado de la tilde (`~`).
- Lo básico de las declaraciones de `if`.

Comandos usados en esta lección:

`echo`
: Referencia a una variable.

`ls`
: Lista el contenido de un directorio.

`readonly`
: Hacer que las variables sean inmutables. Listar todas las variables de sólo lectura en la sesión actual.

`set`
: Enumera todas las variables y funciones de la sesión actual.

`grep`
: Imprime líneas que coincidan con un patrón.

`bash`
: Crear un nuevo shell.

`unset`
: Remover variables.

`export`
: Convierte una variable local en una variable de entorno. Enumera las variables de entorno

`env`
: Enumera las variables de entorno. Ejecuta un programa en un entorno modificado.

`printenv`
: Enumera las variables de entorno. Refiere a una variable.

`chmod`
: Cambiar permisos de un archivo, por ejemplo hacerlo ejecutable.

`history`
: Enumera los comandos anteriores.

`su`
: Cambia la identificación de usuario o lo convierte en superusuario.

`id`
: Imprime la identificación de usuario.

Respuestas a los ejercicios guiados

1. Observe la asignación de la variable en la columna "Comando(s)" e indique si la variable resultante es "Local" o "Global":

Comando(s)	Local	Global
`debian=mother`	Si	No
`ubuntu=deb-based`	Si	No
`mint=ubuntu-based; export mint`	No	Si
`export suse=rpm-based`	No	Si
`zorin=ubuntu-based`	Si	No

2. Estudie el "Comando" y la "Salida" y explica el significado:

Comando	Salida	Signficado
`echo $HISTCONTROL`	`ignoreboth`	Tanto los comandos duplicados como los que empiezan con un espacio no se guardarán en la `historia`.
`echo ~`	`/home/carol`	El `HOME` de `carol` is `/home/carol`.
`echo $DISPLAY`	`reptilium:0:2`	`reptilium` la máquina tiene un servidor X funcionando y estamos usando la segunda pantalla de la pantalla.
`echo $MAILCHECK`	`60`	El correo será revisado cada 60 segundos.
`echo $HISTFILE`	`/home/carol/.bash_history`	`history` se guardará en `/home/carol/.bash_history`.

3. Las variables están siendo puestas incorrectamente en la columna "Comando erróneo". Proporcione la información que falta en "Comando correcto" y "Referencia de la variable" para que obtengamos la "Salida esperada":

Comando erróneo	Comando correcto	Referencia de la variable	Salida esperada
`lizard =chameleon`	`lizard=chameleon`	`echo $lizard`	`chameleon`
`cool lizard=chameleon`	`cool_lizard=chameleon` (Por ejemplo)	`echo $cool_lizard`	`chameleon`
`lizard=cha|me|leon`	`lizard="cha|me|leon"` o `lizard='cha|me|leon'`	`echo $lizard`	`cha|me|leon`
`lizard=/** chameleon **/`	`lizard="/** chameleon **/"` o `lizard='/** chameleon **/'`	`echo "$lizard"`	`/** chamelon **/`
`win_path=C:\path\to\dir\`	`win_path=C:\\path\\to\\dir\\`	`echo $win_path`	`C:\path\to\dir\`

4. Considere el propósito y escriba el comando apropiado:

Propósito	Comando
Establecer el lenguaje del actual shell al español UTF-8 (`es_ES.UTF-8`).	`LANG=es_ES.UTF-8`
Imprime el nombre del directorio actual.	`echo $PWD` or `pwd`
Referencia a la variable de entorno que almacena la información sobre las conexiones `ssh`.	`echo $SSH_CONNECTION`
Establecer el `PATH` para incluir `/home/carol/scripts` como el último directorio para buscar ejecutables.	`PATH=$PATH:/home/carol/scripts`
Establecer el valor de`my_path` en `PATH`.	`my_path=PATH`
Establecer el valor de `my_path` en el de `PATH`.	`my_path=$PATH`

5. Crear una variable local llamada "mammal" y asígnale el valor `gnu`:

```
mammal=gnu
```

6. Usando la sustitución de variables, cree otra variable local llamada `var_sub` con el valor apropiado para que cuando se haga referencia a través de `echo $var_sub` obtengamos: `The value of mammal is gnu`:

```
var_sub="The value of mammal is $mammal"
```

7. Convierte a `mammal` en una variable de entorno:

```
export mammal
```

8. Búscalo con `set` y `grep`:

```
set | grep mammal
```

9. Búscalo con `set` y `grep`:

```
env | grep mammal
```

10. Crear, en dos comandos consecutivos, una variable de entorno llamada `BIRD` cuyo valor es `penguin`:

```
BIRD=penguin; export BIRD
```

11. Crear en dos comandos consecutivos, una variable de entorno llamada `NEW_BIRD` cuyo valor es `yellow-eyed penguin`:

```
export NEW_BIRD="yellow-eyed penguin"
```

o

```
export NEW_BIRD='yellow-eyed penguin'
```

12. Asumiendo que eres `user2`, cree una carpeta llamada `bin` en tu directorio principal:

```
mkdir ~/bin
```

o

```
mkdir /home/user2/bin
```

o

```
mkdir $HOME/bin
```

13. Escriba el comando para agregar la carpeta `~/bin` a su `PATH` para que sea la primera carpeta en la que `bash` busque binarios:

```
PATH="$HOME/bin:$PATH"
```

`PATH=~/bin:$PATH` o `PATH=/home/user2/bin:$PATH` son igualmente válidas.

14. Para garantizar que el valor de `PATH` permanezca inalterado en los reinicios, ¿Qué parte del código — en forma de una declaración `if` — agregarías en el `~/.profile`?

```
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```

Respuestas a los ejercicios de exploración

1. `let`: más que la evaluación de la expresión aritmética:
   - Haz una búsqueda en la página web de `let` y sus implicaciones al establecer las variables y crea una nueva variable local llamada `my_val` cuyo valor es `10` — como resultado de sumar 5 + 5:

     ```
     let "my_val = 5 + 5"
     ```

     o

     ```
     let 'my_val = 5 + 5'
     ```

   - Ahora crea otra variable llamada `your_val`, cuyo valor es `5` — como resultado de dividir el valor de `my_val` entre 2:

     ```
     let "your_val = $my_val / 2"
     ```

     o

     ```
     let 'your_val = $my_val / 2'
     ```

2. ¿El resultado de un comando en una variable? Por supuesto, eso es posible; se llama *sustitución de comandos*. Investiga y estudia la siguiente función llamada `music_info`:

   ```
   music_info(){
   latest_music=`ls -l1t ~/Music | head -n 6`
   echo -e "Your latest 5 music files:\n$latest_music"
   }
   ```

   El resultado del comando `ls -l1t ~/Music | head -n 6` se convierte en el valor de la variable `latest_music`. Luego, se hace referencia a la variable `latest_music` en el comando `echo` (que genera el número total de bytes ocupados por la carpeta `Music` y los últimos cinco archivos de música almacenados en la carpeta `Music` — uno por línea).

   ¿Cuál de los siguientes es un sinónimo válido?

```
latest_music=`ls -l1t ~/Music | head -n 6`
```

Es la opción A:

```
latest_music=$(ls -l1t ~/Music| head -n 6)
```



105.1 Lección 3

Certificación:	LPIC-1
Versión:	5.0
Tema:	105 Shells y scripts
Objetivo:	105.1 Personalizar y usar el entorno de shell
Lección:	3 de 3

Introducción

Después de repasar los shells, los scripts de inicio y las variables, abarcaremos todo el tema de la personalización del shell echando un vistazo a dos elementos muy interesantes: los *alias* y las *funciones*. De hecho, todo el grupo de variables, alias y funciones, es lo que constituye el entorno de shell.

La principal fortaleza tiene que ver con el concepto de encapsulamiento: ofrecen la posibilidad de reunir — bajo un solo comando — una serie de comandos repetitivos o recurrentes.

Creando Alias

Un alias es un nombre sustituto de otro(s) comando(s). Puede ejecutarse como un comando normal, pero en su lugar ejecuta otro comando según la definición de alias

La sintaxis para declarar alias es muy sencilla. Estos se declaran escribiendo la palabra clave `alias` seguida de su asignación. A su vez, dicha asignación consiste en el nombre de alias, un signo igual y uno o más comandos:

```
alias alias_name=command(s)
```

Por ejemplo:

```
$ alias oldshell=sh
```

Este alias iniciará una instancia del shell original `sh` cuando el usuario digite `oldshell` en la terminal:

```
$ oldshell
$
```

El poder de los alias radica en que nos permiten escribir versiones cortas de comandos largos:

```
$ alias ls='ls --color=auto'
```

NOTE Para información sobre `ls` y sus colores, escriba `man dir_colors` en la terminal.

De la misma manera, podemos crear alias para una serie de comandos concatenados — el punto y coma (;) se utiliza como delimitador. Por ejemplo, podemos tener un alias que nos dé información sobre la ubicación del ejecutable `git` y su versión:

```
$ alias git_info='which git;git --version'
```

Para invocar un alias, escribimos su nombre en la terminal:

```
$ git_info
/usr/bin/git
git version 2.7.4
```

El comando `alias` producirá un listado de todos los alias disponibles en el sistema:

```
$ alias
alias git-info='which git;git --version'
alias ls='ls --color=auto'
alias oldshell='sh'
```

El comando `unalias` elimina los alias. Por ejemplo, `unalias git-info`, posteriormente podremos ver como desaparece del listado:

```
$ unalias git-info
$ alias
alias ls='ls --color=auto'
alias oldshell='sh'
```

Como se observa en la sentencia `alias hi='echo We salute you.'`, debemos encerrar los comandos entre comillas (ya sea simple o doble) cuando — los argumentos o parámetros — contienen espacios:

```
$ alias greet='echo Hello world!'
$ greet
Hello world!
```

Los comandos con espacios también son incluidos con esas opciones:

```
$ alias ll='ls -al'
```

Ahora `ll` listará todos los archivos — incluyendo los ocultos (`a`) — en el formato largo (`l`).

Podemos referenciar variables en los alias:

```
$ reptile=uromastyx
$ alias greet='echo Hello $reptile!'
$ greet
Hello uromastyx!
```

La variable también puede ser asignada dentro del alias:

```
$ alias greet='reptile=tortoise; echo Hello $reptile!'
$ greet
Hello tortoise!
```

Podemos escapar de un alias con `\`:


```
$ alias where?='echo $PWD'
$ where?
/home/user2
$ \where?
-bash: where?: command not found
```

Escapar de un alias es útil cuando un alias tiene el mismo nombre que un comando normal. En este caso, el alias tiene prioridad sobre el comando original, sin embargo, este sigue siendo accesible al escapar del alias.

Del mismo modo, podemos colocar un alias dentro de otro alias:

```
$ where?
/home/user2
$ alias my_home=where?
$ my_home
/home/user2
```

Además, también podemos agregar una función dentro de un alias como se muestra a continuación.

Expansión y evaluación de las comillas en los alias

Cuando se usan comillas con variables de entorno, las comillas simples hacen que la expansión sea dinámica:

```
$ alias where?='echo $PWD'
$ where?
/home/user2
$ cd Music
$ where?
/home/user2/Music
```

Sin embargo, con las comillas dobles la expansión se hace de forma estática:

```
$ alias where?="echo $PWD"
$ where?
/home/user2
$ cd Music
$ where?
/home/user2
```

Persistencia de Alias: Scripts de inicio

Al igual que con las variables, para que nuestros alias ganen persistencia, debemos escribirlos en scripts de inicialización que se ejecuten al inicio. Como ya sabemos, un buen archivo para que los usuarios agreguen sus alias personales es `~/.bashrc`. Probablemente encontrarás algunos alias allí (la mayoría de ellos comentados y listos para ser usados eliminando el `#` principal):

```
$ grep alias .bashrc
# enable color support of ls and also add handy aliases
    alias ls='ls --color=auto'
    #alias dir='dir --color=
    #alias vdir='vdir --color=
    #alias grep='grep --color=
    #alias fgrep='fgrep --color'
    #alias egrep='egrep --color=
# some more ls aliases
#ll='ls -al'
#alias la='ls -A'
#alias l='ls -CF'
# ~/.bash_aliases, instead of adding them here directly.
if [ -f ~/.bash_aliases ]; then
   . ~/.bash_aliases
```

Como pueden leer en las últimas tres líneas, se nos ofrece la posibilidad de tener nuestro propio archivo dedicado a los alias — `~/.bash_aliases` — y ser ejecutado por `.bashrc` con cada inicio del sistema. Así que podemos ir por esa opción, crear y editar dicho archivo:

```
###########
# .bash_aliases:
# a file to be populated by the user's personal aliases (and sourced by ~/.bashrc).
###########
alias git_info='which git;git --version'
alias greet='echo Hello world!'
alias ll='ls -al'
alias where?='echo $PWD'
```

Creando funciones

En comparación con los alias, las funciones son más programables y flexibles, especialmente cuando se trata de explotar todo el potencial de las variables incorporadas y los parámetros posicionales de *Bash*. También son muy buenas para trabajar con estructuras de control de flujo, como bucles o condicionales. Podemos pensar en una función como un comando que incluye la lógica a través de bloques o colecciones de otros comandos.

Dos sintaxis para crear funciones

Hay dos sintaxis válidas para definir las funciones.

Usando la palabra clave `function`

Por un lado, podemos usar la palabra clave `function`, seguida del nombre de la función y los comandos entre corchetes:

```
function function_name {
command #1
command #2
command #3
.
.
.
command #n
}
```

Using `()`

Por otro lado, podemos omitir la palabra `function` y usar dos paréntesis justo después del nombre de la función:

```
function_name() {
command #1
command #2
command #3
.
.
.
command #n
}
```

Es común agregar funciones en archivos o scripts. Sin embargo, también se puede escribir con cada comando en el shell prompt, en una línea diferente — note `PS2(>)` que indica una nueva línea después de un salto de línea:

```
$ greet() {
> greeting="Hello world!"
> echo $greeting
> }
```

En cualquier caso, e independientemente de la sintaxis que elijamos, si decidimos saltarnos los saltos de línea y escribir una función en una sola línea, los comandos deben estar separados por punto y coma (también observe el punto y coma que hay después del último comando):

```
$ greet() { greeting="Hello world!"; echo $greeting; }
```

El shell `bash` no se reaccionó negativamente cuando presionamos "Enter", así que nuestra función está lista para ser invocada. Para invocar una función, debemos escribir su nombre en la terminal:

```
$ greet
Hello world!
```

Al igual que con las variables y los alias, si queremos que las funciones sean persistentes a través de los reinicios del sistema tenemos que agregarlas en los scripts de inicialización de shell como `/etc/bash.bashrc` (global) o `~/.bashrc` (local).

WARNING	Después de añadir alias o funciones a cualquier archivo de script de inicio, debe ejecutar tales archivos con `.` o `source` para que los cambios surtan efecto si no quiere salir y volver a entrar o reiniciar el sistema.

Variables especiales incorporadas en Bash

Bourne Again Shell llega con un conjunto de variables especiales que son particularmente útiles para funciones y scripts. Estas son especiales porque sólo pueden ser referenciadas — no asignadas. Aquí hay una lista de las más relevantes:

`$?`

La referencia de esta variable se expande hasta el resultado de la última ejecución del comando. Un valor de `0` significa éxito:

```
$ ps aux | grep bash
user2      420  0.0  0.4  21156  5012 pts/0    Ss   17:10   0:00 -bash
user2      640  0.0  0.0  12784   936 pts/0    S+   18:04   0:00 grep bash
$ echo $?
0
```

Un valor distinto de `0` significa error:

```
user1@debian:~$ ps aux |rep bash
-bash: rep: command not found
user1@debian:~$ echo $?
127
```

`$$`

Se expande hasta el PID del shell (process ID):

```
$ ps aux | grep bash
user2      420  0.0  0.4  21156  5012 pts/0    Ss   17:10   0:00 -bash
user2      640  0.0  0.0  12784   936 pts/0    S+   18:04   0:00 grep bash
$ echo $$
420
```

`$!`

It expands to the PID of the last background job:

```
$ ps aux | grep bash &
[1] 663
$ user2      420  0.0  0.4  21156  5012 pts/0    Ss+  17:10   0:00 -bash
user2      663  0.0  0.0  12784   972 pts/0    S    18:08   0:00 grep bash
^C
[1]+  Done                    ps aux | grep bash
$ echo $!
663
```

NOTE Remember, the ampersand (&) is used to start processes in the background.

Parámetros de posición de `0` a `9`

Se expanden a los parámetros o argumentos que se pasan a la función (alias o script)—`$0` expandiéndose al nombre del script o shell.

Vamos a crear una función para demostrar los parámetros de posición—note `PS2` (>) indicando nuevas líneas después de los saltos de línea:

```
$ special_vars() {
> echo $0
> echo $1
> echo $2
> echo $3
}
```

Ahora, invocaremos la función (`special_vars`) pasándole tres parámetros (`debian`, `ubuntu`, `zorin`):

```
$ special_vars debian ubuntu zorin
-bash
debian
ubuntu
zorin
```

Excelente, funcionó como se esperaba.

WARNING

Aunque pasar parámetros posicionales a los alias es técnicamente posible, no es en absoluto funcional ya que - con los alias - los parámetros posicionales siempre se pasan al final:

```
$ alias great_editor='echo $1 is a great text editor'
$ great_editor emacs
is a great text editor emacs
```

Otras variables especiales incorporadas en Bash incluyen:

`$#`
: Se expande al número de argumentos que se le pasan al comando.

`$@`, `$*`
: Se extienden a los argumentos pasados al comando.

`$_`
: Se expande hasta el último parámetro o el nombre del script (entre otras cosas; ¡revisa "man bash" para conocer más!):

Variables en funciones

Por supuesto, las variables pueden utilizarse dentro de las funciones.

Para probarlo, esta vez crearemos un nuevo archivo vacío llamado `funed` y agregaremos la siguiente función:

```
editors() {

editor=emacs

echo "My editor is: $editor. $editor is a fun text editor."
}
```

Como ya habrán adivinado, debemos obtener el archivo primero para poder invocar la función:

```
$ . funed
```

Y ahora podemos probarlo:

```
$ editors
My editor is emacs. emacs is a fun text editor.
```

Como pueden apreciar, para que la función de `editors` funcione correctamente, la variable `editor` debe ser fijada primero. El alcance de esa variable es local al shell actual y podemos referirnos a ella mientras la sesión este activa:

```
$ echo $editor
emacs
```

Junto con las variables locales también podemos incluir variables de entorno en nuestra función:

```
editors() {

editor=emacs

echo "The text editor of $USER is: $editor."
}

editors
```

Note como esta vez decidimos llamar a la función desde el propio archivo (`editors` en la última línea). De esta manera, cuando hagamos la ejecución del archivo, la función también será invocada — todo a la vez:

```
$ . funed
The text editor of user2 is: emacs.
```

Parámetros de posición en las funciones

Algo similar ocurre con los parámetros posicionales.

Podemos pasarlos a las funciones desde adentro del archivo o el script (note la última línea: `editors tortoise`):

```
editors() {

editor=emacs

echo "The text editor of $USER is: $editor."
echo "Bash is not a $1 shell."
}

editors tortoise
```

Buscamos la fuente del archivo y probamos que funciona:

```
$ . funed
The text editor of user2 is: emacs.
Bash is not a tortoise shell.
```

Y también podemos pasar parámetros posicionales a funciones en la línea de comandos. Para probarlo, nos deshacemos de la última línea del archivo:

```
editors() {

editor=emacs

echo "The text editor of $USER is: $editor."
echo "Bash is not a $1 shell."
}
```

Entonces, tenemos que ejecutar el archivo:

```
$ . funed
```

Finalmente, invocamos la función con `tortoise` como el parámetro posicional `1$` en la línea de comando:

```
$ editors tortoise
The text editor of user2 is: emacs.
Bash is not a tortoise shell.
```

Funciones en Scripts

Las funciones se encuentran principalmente en scripts de Bash.

Convertir nuestro archivo `funed` en un script (lo llamaremos `funed.sh`) es realmente sencillo:

```
#!/bin/bash

editors() {

editor=emacs

echo "The text editor of $USER is: $editor."
echo "Bash is not a $1 shell."
}

editors tortoise
```

¡Eso es! Sólo añadimos dos líneas:

- La primera línea es el *shebang* y define qué programa va a interpretar el script: `#!/bin/bash`. Curiosamente, ese programa es `bash` en sí mismo.
- La última línea es simplemente la invocación de la función.

Ahora sólo queda una cosa — tenemos que hacer el script ejecutable:

```
$ chmod +x funed.sh
```

Y ahora está listo para ser ejecutado:

```
$ ./funed.sh
The text editor of user2 is: emacs.
Bash is not a tortoise shell.
```

NOTE Aprenderás todo sobre el *shell scripting* en las próximas lecciones.

Una función dentro de un Alias

Como se ha dicho antes, podemos agregar una función dentro de un alias:

```
$ alias great_editor='gr8_ed() { echo $1 is a great text editor; unset -f gr8_ed;
}; gr8_ed'
```

Este largo valor del alias merece una explicación. Vamos a desglosarlo:

- Primero está la función en sí misma: `gr8_ed() { echo $1 is a great text editor; unset -f gr8_ed; }`
- El último comando de la función — `unset -f gr8_ed` — remueve la función para que no permanezca en la sesión actual de `bash` después de que el alias sea llamado.
- Por último, pero no menos importante, para tener una invocación de alias exitosa, también debemos invocar la función: `gr8_ed`.

Invoquemos el alias y probemos si funciona:

```
$ great_editor emacs
emacs is a great text editor
```

Como se muestra en `unset -f gr8_ed`, el comando `unset` no sólo se usa para remover variables, sino también funciones. De hecho, hay opciones específicas:

`unset -v`
: para variables

`unset -f`
: para funciones

Si se usa sin opciones, `unset` tratará de desajustar una variable primero y — si falla — entonces tratará de desajustar una función.

Una función dentro de un Alias

Ahora mencionamos que queremos comunicar dos cosas a `user2` cada vez que se registre en el sistema:

- Saludar y recomendar un editor de texto.
- Ya que está empezando a agregar muchos archivos de video de `Matrosk`" en su carpeta `$HOME/Video`, también queremos darle una advertencia.

Para lograr ese propósito, hemos puesto las siguientes dos funciones en `/home/user2/.bashrc`:

La primera función (`check_vids`) hace el chequeo de los archivos `.mkv` y la advertencia:

```
check_vids() {
    ls -1 ~/Video/*.mkv > /dev/null 2>&1
    if [ "$?" = "0" ];then
        echo -e "Remember, you must not keep more than 5 video files in your Video
folder.\nThanks."
    else
    echo -e "You do not have any videos in the Video folder. You can keep up to
5.\nThanks."
    fi
}
```

`check_vids` hace tres cosas:

- Lista los archivos `mkv` en `~/Video` enviando la salida — y cualquier error — al llamado *bit-bucket* (`/dev/null`).
- Prueba la salida del comando anterior para el éxito.
- Dependiendo del resultado de la prueba, imprime uno de los dos mensajes.

La segunda función es una versión modificada de nuestra función de `editors`:

```
editors() {

editor=emacs

echo "Hi, $USER!"
echo "$editor is more than a text editor!"

check_vids
}

editors
```

Es importante observar dos cosas:

- El último comando de `editors` invoca `check_vids` para que ambas funciones se encadenen: El saludo, el elogio, el chequeo y la advertencia se ejecutan en secuencia.
- `editors` es el mismo punto de entrada a la secuencia de funciones, por lo que se invoca en la última línea (`editors`).

Ahora, entremos como `user2` y probemos que funciona:

```
# su - user2
Hi, user2!
emacs is more than a text editor!
Remember, you must not keep more than 5 video files in your Video folder.
Thanks.
```

Ejercicios guiados

1. Completa la tabla con `Sí` o `No` considerando las capacidades de los alias y las funciones:

Características	Alias	Funciones
Las variables locales pueden ser usadas		
Las variables de entorno pueden ser utilizadas		
Se puede escapar con `\`		
Puede ser recursivo.		
Muy productivo cuando se usa con parámetros posicionales		

2. Introduzca el comando que lista todos los alias en su sistema:

3. Escribe un alias llamado `logg` que enumere todos los archivos `ogg` en `~/Music` — uno por línea:

4. Invoca el alias para probar que funciona:

5. Ahora, modifica el alias para que imprima el usuario de la sesión y dos puntos antes del listado:

6. Invóquelo de nuevo para probar que esta nueva versión también funciona:

7. Enumera todos los alias de nuevo y comprueba que tu alias de `logg` aparece en el listado:

8. Quite el alias:

9. Estudie las columnas `Nombre del alias`,`Comando(s) aliado(s)` y asigne los alias a sus valores correctamente:

Nombre del alias	Comando(s) aliado(s)	Asignación de Alias
`b`	`bash`	
`bash_info`	`which bash` + `echo "$BASH_VERSION"`	
`kernel_info`	`uname -r`	
`greet`	`echo Hi, $USER!`	
`computer`	`pc=slimbook` + `echo My computer is a $pc`	

10. Como `root`, escribe una función llamada `my_fun` en `etc/bash.bashrc`. La función debe saludar al usuario y decirle cuál es su PATH. Invóque la función para que el usuario reciba los dos mensajes cada vez que inicie sesión:

11. Ingrese como `user2` para comprobar que funciona:

12. Escriba la misma función en una sola línea:

13. Invoca la función:

14. Desconecta la función:

15. Esta es una versión modificada de la función `special_vars`:

```
$ special_vars2() {
> echo $#
> echo $_
> echo $1
> echo $4
> echo $6
> echo $7
> echo $_
> echo $@
> echo $?
> }
```

Este es el comando que usamos para invocarlo:

```
$ special_vars2 crying cockles and mussels alive alive oh
```

Adivina las salidas:

Referencia	Valor
`echo $#`	
`echo $_`	
`echo $1`	
`echo $4`	
`echo $6`	
`echo $7`	
`echo $_`	
`echo $@`	
`echo $?`	

16. Basándonos en la función de muestra (`check_vids`) en la sección “Función dentro de una función”, escribe una función llamada `check_music` para incluirla en un script de inicio `bash` que acepte parámetros posicionales para que podamos modificarla fácilmente:
    - El tipo de archivo que se comprueba: `ogg`
    - El directorio en el que se guardan los archivos: `~/Music`
    - El tipo de archivo que se guarda: `music`
    - El número de archivos que se guardan: `7`

Ejercicios de exploración

1. Las funciones de sólo lectura son aquellas cuyo contenido no podemos modificar. Haga una investigación sobre *funciones de sólo lectura* y complete la siguiente tabla:

Nombre de la función	Haz que sea sólo de lectura	Lista de todas las funciones de sólo lectura
`my_fun`		

2. Busca en la web cómo modificar `PS1` y cualquier otra cosa que necesites para escribir una función llamada `fyi` (que se colocará en un script de inicio) que le da al usuario la siguiente información:
   - Nombre del usuario
   - Directorio principal
   - Nombre del host
   - Tipo de sistema operativo
   - Buscar la ruta (PATH) de ejecutables
   - Directorio de correo
   - Con qué frecuencia se revisa el correo
   - ¿Cuántos shells tiene la sesión actual?
   - Prompt (deberías modificarlo para que muestre `<user>@<host-date>`)

Resumen

En esta lección aprendió:

- Tanto los alias como las funciones son características importantes de shell que nos permiten encapsular bloques de código recurrentes.
- Los alias son útiles para tener versiones más cortas de comandos largos y/o complicados.
- Las funciones son procedimientos que implementan la lógica y nos permiten automatizar tareas, especialmente cuando se usan en scripts.
- La sintaxis para escribir alias y funciones.
- Como concatenar varios comandos por medio del punto y coma (`;`).
- Como usar correctamente las comillas con los alias.
- Como hacer que los alias y las funciones sean persistentes.
- Variables Bash especiales incorporadas: `$?`, `$$`, `$!`, parámetros de posición (`$0`-`$9`), `$#`, `$@`, `$*` y `$_`.
- Como usar variables y parámetros posicionales con las funciones.
- Como usar funciones en los scripts.
- Como invocar una función desde un alias.
- Como invocar una función desde otra función.
- Lo básico para crear un script `bash`.

Comandos y palabras clave utilizadas en esta lección:

`alias`
: Crear alias.

`unalias`
: Eliminar alias.

`cd`
: Cambiar de directorio.

`grep`
: Imprime líneas que coincidan con un patrón.

`function`
: Palabra clave de Shell para crear funciones

`.`
: Ejecutar un archivo

`source`
: Realiza una ejecución de un archivo con el comando source

`ps`
: Captura de los procesos actuales.

`echo`
: Imprimir en pantalla una línea de texto.

`chmod`
: Cambiar permisos de un archivo, por ejemplo hacerlo ejecutable.

`unset`
: Remover variables and funciones.

`su`
: Cambia la identificación de usuario o lo convierte en superusuario.

Respuesta a los ejercicios guiados

1. Completa la tabla con `Sí` o `No` considerando las capacidades de los alias y las funciones:

Características	Alias	Funciones
Las variables locales pueden ser usadas	Sí	Sí
Las variables de entorno pueden ser utilizadas	Sí	Sí
Se puede escapar con `\`	Sí	No
Puede ser recursivo.	Sí	Sí
Muy productivo cuando se usa con parámetros posicionales	No	Sí

2. Introduzca el comando que lista todos los alias en su sistema:

```
alias
```

3. Escribe un alias llamado `logg` que enumere todos los archivos `ogg` en `~/Music` — uno por línea:

```
alias logg='ls -1 ~/Music/*ogg'
```

4. Invoca el alias para probar que funciona:

```
logg
```

5. Ahora, modifica el alias para que imprima el usuario de la sesión y dos puntos antes del listado:

```
alias logg='echo $USER:; ls -1 ~/Music/*ogg'
```

6. Invóquelo de nuevo para probar que esta nueva versión también funciona:

```
logg
```

7. Enumera todos los alias de nuevo y comprueba que tu alias de `logg` aparece en el listado:

```
alias
```

8. Quite el alias:

```
unalias logg
```

9. Estudie las columnas `Nombre del alias`,`Comando(s) aliado(s)` y asigne los alias a sus valores correctamente:

Nombre del alias	Comando(s) aliado(s)	Asignación de Alias
`b`	`bash`	`alias b=bash`
`bash_info`	`which bash` + `echo "$BASH_VERSION"`	`alias bash_info='which bash; echo "$BASH_VERSION"'`
`kernel_info`	`uname -r`	`alias kernel_info='uname -r'`
`greet`	`echo Hi, $USER!`	`alias greet='echo Hi, $USER'`
`computer`	`pc=slimbook` + `echo My computer is a $pc`	`alias computer='pc=slimbook; echo My computer is a $pc'`

NOTE Las comillas simples también pueden ser reemplazadas por dobles.

10. Como `root`, escribe una función llamada `my_fun` en `etc/bash.bashrc`. La función debe saludar al usuario y decirle cuál es su PATH. Invóque la función para que el usuario reciba los dos mensajes cada vez que inicie sesión:

Opción A:

```
my_fun() {
echo Hello, $USER!
echo Your path is: $PATH
}
my_fun
```

Opción B:

```
function my_fun {
echo Hello, $USER!
echo Your path is: $PATH
}
my_fun
```

11. Ingrese como `user2` para comprobar que funciona:

```
su - user2
```

12. Escriba la misma función en una sola línea:

Opción A:

```
my_fun() { echo "Hello, $USER!"; echo "Your path is: $PATH"; }
```

Opción B:

```
function my_fun { echo "Hello, $USER!"; echo "Your path is: $PATH"; }
```

13. Invoca la función:

```
my_fun
```

14. Remover la función:

```
unset -f my_fun
```

15. Esta es una versión modificada de la función `special_vars`:

```
$ special_vars2() {
> echo $#
> echo $_
> echo $1
> echo $4
> echo $6
> echo $7
> echo $_
> echo $@
> echo $?
> }
```

Este es el comando que usamos para invocarlo:

```
$ special_vars2 crying cockles and mussels alive alive oh
```

Adivina las salidas:

Referencia	Valor
`echo $#`	`7`
`echo $_`	`7`
`echo $1`	`crying`
`echo $4`	`mussels`
`echo $6`	`alive`
`echo $7`	`oh`
`echo $_`	`oh`
`echo $@`	`crying cockles and mussels alive alive oh`
`echo $?`	`0`

16. Basándonos en la función de muestra (`check_vids`) en la sección “Función dentro de una función”, escribe una función llamada `check_music` para incluirla en un script de inicio `bash` que acepte parámetros posicionales para que podamos modificarla fácilmente:
    - El tipo de archivo que se comprueba: `ogg`
    - El directorio en el que se guardan los archivos: `~/Music`
    - El tipo de archivo que se guarda: `music`
    - El número de archivos que se guardan: `7`

```
check_music() {
    ls -1 ~/$1/*.$2 > ~/.mkv.log 2>&1
    if [ "$?" = "0" ];then
        echo -e "Remember, you must not keep more than $3 $4 files in your
$1 folder.\nThanks."
    else
     echo -e "You do not have any $4 files in the $1 folder. You can keep
up to $3.\nThanks."
    fi
}

check_music Music ogg 7 music
```

Respuestas a los ejercicios de exploración

1. Las funciones de sólo lectura son aquellas cuyo contenido no podemos modificar. Haga una investigación sobre *funciones de sólo lectura* y complete la siguiente tabla:

Nombre de la función	Haz que sea sólo de lectura	Lista de todas las funciones de sólo lectura
`my_fun`	`readonly -f my_fun`	`readonly -f`

2. Busca en la web cómo modificar `PS1` y cualquier otra cosa que necesites para escribir una función llamada `fyi` (que se colocará en un script de inicio) que le da al usuario la siguiente información:
   - Nombre del usuario
   - Directorio principal
   - Nombre del host
   - Tipo de sistema operativo
   - Buscar la ruta (PATH) de ejecutables
   - Directorio de correo
   - Con qué frecuencia se revisa el correo
   - ¿Cuántos shells tiene la sesión actual? *
   - prompt (deberías modificarlo para que muestre `<user>@<host-date>`)

```
fyi() {
    echo -e "For your Information:\n
    Username: $USER
    Home directory: $HOME
    Host: $HOSTNAME
    Operating System: $OSTYPE
    Path for executable files: $PATH
    Your mail directory is $MAIL and is searched every $MAILCHECK seconds.
    The current level of your shell is: $SHLVL"
    PS1="\u@\h-\d "
}

fyi
```



105.2 Personalizar y escribir scripts sencillos

Referencia al objetivo del LPI

LPIC-1 5.0, Exam 102, Objective 105.2

Importancia

4

Áreas de conocimiento clave

- Usar la sintaxis estándar sh (bucles, tests).
- Usar la sustitución de comandos.
- Evaluar correctamente el código de retorno de un comando en caso de éxito, fracaso o cualquier otra información que proporcione la salida del comando.
- Ejecutar comandos en cadena.
- Realizar envío de correo condicional al superusuario.
- Seleccionar correctamente el intérprete del script mediante la línea inicial o shebang (`#!`).
- Gestionar la ubicación, los propietarios, la ejecución y los permisos suid de los scripts.

Lista parcial de archivos, términos y utilidades

- `for`
- `while`
- `test`
- `if`
- `read`
- `seq`

- `exec`
- `||`
- `&&`



105.2 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	105 Shells y scripts
Objetivo:	105.2 Personalizar y escribir scripts sencillos
Lección:	1 de 2

Introducción

El entorno del shell de Linux permite el uso de archivos — llamados *scripts* — que contienen comandos de cualquier programa disponible en el sistema, combinado con comandos en shell para automatizar las tareas personalizadas de un usuario y/o un sistema. De hecho, muchas de las tareas de mantenimiento del sistema operativo son realizadas por scripts que consisten en secuencias de comandos, estructuras de decisión y bucles condicionales. Aunque la mayoría de las veces los scripts están destinados a tareas relacionadas con el propio sistema operativo, también son útiles para tareas orientadas al usuario, como el renombramiento masivo de archivos, la recopilación y el análisis sintáctico de datos o cualquier otra actividad repetitiva de la línea de comandos. Los scripts no son más que archivos de texto que se comportan como programas. En un programa — el intérprete — lee y ejecuta las instrucciones que aparecen en el script. El intérprete también puede iniciar una sesión interactiva donde los comandos — incluyendo los scripts — se leen y ejecutan a medida que se introducen, como es el caso de las sesiones de shell de Linux. Los archivos de script pueden agrupar esas instrucciones y comandos cuando se vuelven demasiado complejos para ser implementados como un alias o una función de shell personalizada. Además, los archivos de script pueden ser mantenidos como los programas convencionales y, al ser sólo archivos de texto, pueden ser creados y modificados con cualquier editor de texto simple.

Estructura y ejecución del script

Básicamente, un archivo de script es una secuencia ordenada de comandos que debe ser ejecutada por un intérprete de comandos correspondiente. La forma en que un intérprete lee un archivo de script varía y hay distintas maneras de hacerlo en una sesión de shell Bash, pero el intérprete por defecto de un archivo de script será el indicado en la primera línea del script, justo después de los caracteres `#!` (conocido como *shebang*). En un script con instrucciones para el shell Bash, la primera línea debe ser `#!/bin/bash`. Al indicar esta línea, el intérprete de todas las instrucciones del archivo será `/bin/bash`. Excepto la primera línea, todas las demás líneas que empiezan con el carácter de hash `#` serán ignoradas, así que pueden ser usadas para colocar recordatorios y comentarios. Las líneas en blanco también se ignoran. Por lo tanto, un archivo de script de shell muy conciso puede ser escrito de la siguiente manera:

```
#!/bin/bash

# A very simple script

echo "Cheers from the script file! Current time is: "

date +%H:%M
```

Este script sólo tiene dos instrucciones para el intérprete de `/bin/bash`: el comando incorporado `echo` y el comando `date`. La forma más básica de ejecutar un archivo de script es ejecutar el intérprete con la ruta del script como argumento. Así que, asumiendo que el ejemplo anterior fue guardado en un archivo de script llamado `script.sh` en el directorio actual, será leído e interpretado por Bash con el siguiente comando:

```
$ bash script.sh
Cheers from the script file! Current time is:
10:57
```

El comando `echo` añadirá automáticamente una nueva línea después de mostrar el contenido, pero la opción `-n` suprimirá este comportamiento. Por lo tanto, el uso de `echo -n` en el script hará que la salida de ambos comandos aparezca en la misma línea:

```
$ bash script.sh
Cheers from the script file! Current time is: 10:57
```

Aunque no es obligatorio, el sufijo `.sh` ayuda a identificar los scripts de shell a la hora de listarlos o

buscarlos.

TIP	Bash llamará a cualquier comando que se indique después de la `#!` como el intérprete del archivo del script. Puede ser útil, por ejemplo, emplear el shebang para otros lenguajes de scripts, como *Python* (`#!/usr/bin/python`), *Perl* (`#!/usr/bin/perl`) o *awk* (`#!/usr/bin/awk`).

Si el archivo de escritura está destinado a ser ejecutado por otros usuarios del sistema, es importante comprobar si se han establecido los permisos de lectura adecuados. El comando `chmod o+r script.sh` dará permiso de lectura a todos los usuarios del sistema, permitiéndoles ejecutar `script.sh` colocando la ruta del archivo como argumento del comando `bash`. Alternativamente, el script puede tener el permiso de ejecución para que el archivo pueda ser ejecutado como un comando convencional. El bit de ejecución se activa en el archivo script con el comando `chmod`:

```
$ chmod +x script.sh
```

Con el bit de ejecución activado, el archivo de script llamado `script.sh` en el directorio actual puede ser ejecutado directamente con el comando `./script.sh`. Los scripts ubicados en un directorio listado en la variable de entorno `PATH` también serán accesibles sin su ruta completa.

WARNING	Un script que realiza acciones restringidas puede tener su permiso SUID activado, por lo que los usuarios normales también pueden ejecutar el script con privilegios de root. En este caso, es muy importante asegurarse de que ningún usuario que no sea root tenga el permiso para escribir en el archivo. De lo contrario, un usuario ordinario podría modificar el archivo para realizar operaciones arbitrarias y potencialmente dañinas.

La colocación y la indentación de los comandos en los archivos de escritura no son demasiado rígidos. Cada línea de un script se ejecutará como un comando de shell ordinario. En la misma secuencia en que la línea aparece en el archivo de script, y las mismas reglas que se aplican al prompt, también se aplica a cada línea del script de manera individual. Es posible colocar dos o más comandos en la misma línea, separados por punto y coma:

```
echo "Cheers from the script file! Current time is:" ; date +%H:%M
```

Aunque este formato puede ser conveniente a veces, su uso es opcional, ya que los comandos secuenciales pueden colocarse un comando por línea y se ejecutarán tal como estaban separados por punto y coma. En otras palabras, el punto y coma puede ser reemplazado por un nuevo carácter de

línea en los script Bash.

Cuando se ejecuta un script, los comandos contenidos en este no se ejecutan directamente en la sesión actual, sino que se ejecutan mediante un nuevo proceso Bash, llamado *sub-shell*. Este evita que el script sobrescriba las variables de entorno de la sesión actual y que deje modificaciones desatendidas en la sesión actual. Si el objetivo es ejecutar el contenido del script en la sesión de shell actual, entonces debe ser ejecutado con `source script.sh` o `. script.sh` (note que hay un espacio entre el punto y el nombre del script).

Como sucede con la ejecución de cualquier otro comando, el prompt del shell sólo estará disponible de nuevo cuando el script termine su ejecución y su código (estado) de salida estará disponible en la variable `$?`. Para cambiar este comportamiento, de modo que el shell actual también termine cuando el script lo haga, el script — o cualquier otro comando — puede ser precedido por el comando `exec`. Este comando también reemplazará el código de estado de salida de la sesión actual del shell por el suyo propio.

Variables

Las variables en los shell scripts se comportan de la misma manera que en las sesiones interactivas, dado que el intérprete es el mismo. Por ejemplo, el formato `SOLUTION=42` (sin espacios alrededor del signo igual) asignará el valor `42` a la variable denominada `SOLUTION`. Por convención, las letras mayúsculas se usan para los nombres de las variables, pero no es obligatorio. Los nombres de las variables no pueden, sin embargo, comenzar con caracteres no alfabéticos.

Además de las variables ordinarias creadas por el usuario; los scripts de Bash también tienen un conjunto de variables especiales llamadas *parámetros*. A diferencia de las variables ordinarias, los nombres de los parámetros comienzan con un carácter no alfabético que designa su función. Los argumentos que se pasan a un guión y otra información útil se almacenan en parámetros como `$0`, `$*`, `$?`, etc., donde el carácter que sigue al signo del dólar indica la información que se debe buscar:

`$*`

Todos los argumentos pasaron al script.

`$@`

Todos los argumentos pasados al script. Si se usa con comillas dobles como en `"$@"`, todos los argumentos serán encerrados entre comillas dobles.

`$#`

El número de argumentos

`$0`

El nombre del script.

`$!`

PID del último programa ejecutado

`$$`

PID del shell actual.

`$?`

El status (código) del último comando terminado. En los procesos POSIX, un valor numérico de `0` significa que el último comando se ejecutó con éxito, lo que también se aplica a los scripts de shell.

Un *parámetro posicional* es un parámetro denotado por uno o más dígitos, aparte del dígito `0`. Por ejemplo, la variable `$1` corresponde al primer argumento dado al script (parámetro posicional uno), `$2` corresponde al segundo argumento, y así sucesivamente. Si la posición de un parámetro es mayor que nueve, debe ser referenciada con llaves, como `${10}`, `${11}`, etc.

En cambio, las variables ordinarias están destinadas a almacenar valores insertados manualmente o la salida generada por otros comandos. Por ejemplo, el comando "read", puede ser usado dentro del script para pedirle al usuario que introduzca datos durante la ejecución del mismo:

```
echo "Do you want to continue (y/n)?"
read ANSWER
```

El valor devuelto se almacenará en la variable `ANSWER`. Si no se suministra el nombre de la variable, se usará por defecto el nombre de la variable `REPLY`. También es posible usar el comando `read` para leer más de una variable simultáneamente:

```
echo "Type your first name and last name:"
read NAME SURNAME
```

En este caso, cada término separado del espacio será asignado a las variables `NAME` y `SURNAME` respectivamente. Si el número de términos dados es mayor que el número de variables, los términos excedentes se almacenarán en la última variable. Inclusive `read` puede mostrar el mensaje al usuario con la opción `-p`, haciendo que el comando `echo` sea redundante en este caso:

```
read -p "Type your first name and last name:" NAME SURNAME
```

Los scripts que realizan tareas del sistema, a menudo requieren información proporcionada por otros programas. La *notación backtick* puede usarse para almacenar la salida de un comando en una variable:

```
$ OS=`uname -o`
```

En el ejemplo, la salida del comando "uname -o" se almacenará en la variable `OS`. Un resultado idéntico se producirá con "$()`:

```
$ OS=$(uname -o)
```

La longitud de una variable, es decir, la cantidad de caracteres que contiene, se devuelve preparando un hash `#` antes del nombre de la variable. Esta característica, requiere el uso de las llaves para indicar la variable:

```
$ OS=$(uname -o)
$ echo $OS
GNU/Linux
$ echo ${#OS}
9
```

Bash también cuenta con variables de matriz unidimensional, por lo que un conjunto de elementos relacionados puede ser almacenado con un solo nombre de variable. Cada elemento de una matriz tiene un índice numérico, que debe utilizarse para escribir y leer valores en el elemento correspondiente. A diferencia de las variables ordinarias, las matrices deben ser declaradas con el comando incorporado en Bash `declare`. Por ejemplo, para declarar una variable llamada `SIZES` como una matriz:

```
$ declare -a SIZES
```

Las matrices también pueden declararse implícitamente partir de una lista predefinida de elementos, utilizando la notación de paréntesis:

```
$ SIZES=( 1048576 1073741824 )
```

En el ejemplo, los dos valores enteros se almacenaron en la matriz `SIZES`. Se debe hacer referencia a los elementos de la matriz mediante llaves y corchetes. De lo contrario, Bash no cambiará ni mostrará el elemento correctamente. Como los índices de matriz comienzan en 0, el contenido del primer elemento está en `${SIZES[0]}`, el segundo elemento está en `${SIZES[1]}`, y así sucesivamente:

```
$ echo ${SIZES[0]}
1048576
$ echo ${SIZES[1]}
1073741824
```

A diferencia de la lectura, el cambio del contenido de un elemento de la matriz se realiza sin las llaves (Por ejemplo, `SIZES[0]=1048576`). Al igual que con las variables ordinarias, la longitud de un elemento en una matriz se devuelve con el carácter númeral (por ejemplo, `${#SIZES[0]}` para la longitud del primer elemento en la matriz `SIZES`). Se devuelve el número total de elementos en una matriz si se usa `@` o `*` como índice:

```
$ echo ${#SIZES[@]}
2
$ echo ${#SIZES[*]}
2
```

También se pueden declarar las matrices utilizando la salida de un comando como elementos iniciales mediante la sustitución de comandos. El siguiente ejemplo muestra cómo crear una matriz Bash cuyos elementos son los sistemas de archivos soportados por el sistema actual:

```
$ FS=( $(cut -f 2 < /proc/filesystems) )
```

El comando `cut -f 2 < /proc/filesystems` mostrará todos los sistemas de archivos actualmente soportados por el kernel en ejecución (como se indica en la segunda columna del archivo `/proc/filesystems`), por lo que el arreglo `FS` ahora contiene un elemento para cada sistema de archivos soportado. Se puede utilizar cualquier contenido de texto para inicializar un arreglo ya que, por defecto, cualquier término delimitado por los caracteres *espacio*, *tabulación* o *nueva línea (newline)* se convertirá en un elemento del arreglo.

TIP Bash trata cada carácter de una variable de entorno `$IFS` (*Input Field Separator*) como un delimitador. Para cambiar el delimitador de campo a caracteres de nueva línea, la variable IFS debe ser reseteada con el comando `IFS=$'\n'`.

Expresiones aritméticas

Bash proporciona un método práctico para realizar operaciones aritméticas enteras con el comando incorporado `expr`. Dos variables numéricas, `$VAL1` y `$VAL2` por ejemplo, se pueden sumar con el siguiente comando:

```
$ SUM=`expr $VAL1 + $VAL2`
```

El resultado del ejemplo estará disponible en la variable `$SUM`. El comando `expr` puede ser reemplazado por `$(())`, así que el ejemplo anterior puede ser reescrito como `SUM=$(($VAL1+$VAL2))`. Las expresiones de poder también están permitidas con el operador de doble asterisco, por lo que la declaración anterior del arreglo `SIZES=(1048576 1073741824)` puede ser reescrita como `SIZES=($((1024**2)) $((1024**3)) )`.

La sustitución de comandos también se puede utilizar en las expresiones aritméticas. Por ejemplo, el archivo `/proc/meminfo` tiene información detallada sobre la memoria del sistema, incluyendo el número de bytes libres en la RAM:

```
$ FREE=$(( 1000 * `sed -nre '2s/[^[:digit:]]//gp' < /proc/meminfo` ))
```

El ejemplo muestra cómo el comando `sed` puede ser usado para analizar el contenido de `/proc/meminfo` dentro de la expresión aritmética. La segunda línea del archivo `/proc/meminfo` contiene la cantidad de memoria libre en miles de bytes, así que la expresión aritmética la multiplica por 1000 para obtener el número de bytes libres en la RAM.

Ejecución condicional

Algunos scripts no suelen estar destinados a ejecutar todos los comandos de su contenido, sino sólo aquellos comandos que coinciden con un criterio predefinido. Por ejemplo, un script de mantenimiento puede enviar un mensaje de advertencia al correo electrónico del administrador sólo si la ejecución de un comando falla. Bash proporciona métodos específicos para evaluar el éxito de la ejecución del comando y estructuras condicionales generales, más similares a los que se encuentran en los lenguajes de programación más populares.

Al separar los comandos con `&&`, el comando de la derecha se ejecutará sólo si el comando de la izquierda no encontró un error, es decir, si su estado de salida era igual a `0`:

```
COMMAND A && COMMAND B && COMMAND C
```


El comportamiento opuesto ocurre si los comandos se separan con `||`. En este caso, el siguiente comando se ejecutará sólo si el comando anterior encontró un error, es decir, si su código de estado retornó un valor difernte a 0.

Una de las características más importantes de todos los lenguajes de programación es la capacidad de ejecutar comandos dependiendo de condiciones previamente definidas. La forma más sencilla de ejecutar comandos condicionalmente es utilizar el comando incorporado en Bash `if`, que ejecuta uno o más comandos sólo si el comando dado como argumento devuelve un código de estado "0" (éxito). Otro comando, `test` puede ser usado para evaluar muchos criterios especiales, por lo que se usa mayormente en conjunto con `if`. En el siguiente ejemplo, el mensaje `Confirmed: /bin/bash is executable.` se mostrará si el archivo `bin/bash` existe y es ejecutable:

```
if test -x /bin/bash ; then
  echo "Confirmed: /bin/bash is executable."
fi
```

La opción `-x` hace que el comando `test` devuelva un código de estado "0" sólo si la ruta dada es un archivo ejecutable. El siguiente ejemplo muestra otra forma de conseguir exactamente el mismo resultado, ya que los corchetes pueden utilizarse como sustituto de`test`:

```
if [ -x /bin/bash ] ; then
  echo "Confirmed: /bin/bash is executable."
fi
```

La instrucción `else` es opcional a la estructura `if` y puede, si está presente, definir un comando o secuencia de comandos a ejecutar si la expresión condicional no es verdadera:

```
if [ -x /bin/bash ] ; then
  echo "Confirmed: /bin/bash is executable."
else
  echo "No, /bin/bash is not executable."
fi
```

Las estructuras `if` siempre deben terminar con `fi`, así el intérprete de Bash sabe donde terminan los comandos condicionales.

Salidas de un Script

Incluso cuando la finalidad de una secuencia de comandos sólo implica operaciones orientadas a

archivos, es importante mostrar mensajes relacionados con el progreso en la salida estándar, de modo que el usuario se mantenga informado de cualquier problema y pueda eventualmente utilizar esos mensajes para generar registros de operaciones.

El comando incorporado de Bash `echo` se utiliza comúnmente para mostrar cadenas simples de texto, pero también proporciona algunas características extendidas. Con la opción `-e`, el comando `echo` es capaz de mostrar caracteres especiales usando secuencias de escape (una secuencia de barra invertida que designa un carácter especial). Por ejemplo:

```
#!/bin/bash

# Obtiene el nombre genérico del sistema operativo
OS=$(uname -o)

# Obtiene la cantidad de memoria libre en bytes
FREE=$(( 1000 * `sed -nre '2s/[^[:digit:]]//gp' < /proc/meminfo` ))

echo -e "Operating system:\t$OS"
echo -e "Unallocated RAM:\t$(( $FREE / 1024**2 )) MB"
```

Mientras que el uso de comillas es opcional cuando se utiliza `echo` sin opciones, es necesario añadirlas cuando se utiliza la opción `-e`, de lo contrario los caracteres especiales no se renderizarán correctamente. En el script anterior, ambos comandos `echo` utilizan el carácter de tabulación `\t` para alinear el texto, lo que resulta en la siguiente salida:

```
Operating system:       GNU/Linux
Unallocated RAM:        1491 MB
```

El carácter de nueva línea `\n` puede ser usado para separar las líneas de salida, así que la misma salida se obtiene combinando los dos comandos `echo` en uno solo:

```
echo -e "Operating system:\t$OS\nUnallocated RAM:\t$(( $FREE / 1024**2 )) MB"
```

Aunque es apto para mostrar la mayoría de los mensajes de texto, el comando `echo` puede no ser adecuado para mostrar patrones de texto más específicos. El comando `printf` incorporado en la barra de herramientas, brinda más control sobre cómo mostrar las variables. El comando `printf` utiliza el primer argumento como formato de la salida, donde los marcadores de posición serán reemplazados por los siguientes argumentos en el orden en que aparecen en la línea de comandos. Por ejemplo, el mensaje del ejemplo anterior podría generarse con el siguiente comando `printf`:

```
printf "Operating system:\t%s\nUnallocated RAM:\t%d MB\n" $OS $(( $FREE / 1024**2
))
```

El marcador de posición `%s` está destinado al contenido del texto (será reemplazado por la variable `$OS`) y el marcador de posición `%d` está destinado a los números enteros (será reemplazado por el número resultante de megabytes libres en la RAM). El comando`printf` no añade un carácter de nueva línea al final del texto, por lo que el carácter de nueva línea `\n` debe ser colocado al final del patrón si es necesario. Todo el patrón debe ser interpretado como un único argumento, por lo que debe ser incluido entre comillas.

TIP El formato de sustitución realizado por `printf` puede personalizarse utilizando el mismo formato utilizado por la función `printf` del lenguaje de programación C. La referencia completa de la función `printf` se puede encontrar en su página de `man`, a la que se accede con el comando `man 3 printf`.

Con `printf`, las variables se colocan fuera del patrón de texto, lo que permite almacenar el patrón de texto en una variable separada:

```
MSG='Operating system:\t%s\nUnallocated RAM:\t%d MB\n'
printf "$MSG" $OS $(( $FREE / 1024**2 ))
```

Este método es particularmente útil para mostrar distintos formatos de salida según las necesidades del usuario. Por ejemplo, facilita la escritura de un script que utilice un patrón de texto distinto, como puede ser una lista CSV (*Comma Separated Values*) en lugar de un mensaje de salida predeterminado.

Ejercicios guiados

1. La opción `-s` del comando `read` es útil para introducir contraseñas, ya que no mostrará el contenido que se está escribiendo en la pantalla. ¿Cómo podría usarse este comando para almacenar la entrada del usuario en la variable `PASSWORD` mientras se oculta el contenido escrito?

2. El único propósito del comando `whoami` es mostrar el nombre de usuario que lo ha llamado, por lo que se utiliza principalmente dentro de los scripts para identificar al usuario que lo está ejecutando. Dentro de un script Bash, ¿cómo podría la salida del comando `whoami` ser almacenada en la variable llamada `WHO`?

3. ¿Qué operador de Bash debería estar entre los comandos `apt-get dist-upgrade` y `systemctl reboot`, si el usuario root quisiera ejecutar este último y solo si el comando `apt-get dist-upgrade` haya terminado con éxito?

Ejercicios de exploración

1. Después de intentar ejecutar un script Bash recién creado, el usuario recibe el siguiente mensaje de error:

   ```
   bash: ./script.sh: Permission denied
   ```

   Considerando que el archivo `./script.sh` fue creado por el mismo usuario, ¿cuál sería la causa probable de este error?

2. Supongamos que un script llamado `do.sh` es ejecutable y el enlace simbólico llamado `do.sh` apunta a este. Dentro del script, ¿cómo podrías identificar si el nombre del archivo de llamada era `do.sh` o `undo.sh`?

3. En un sistema con un servicio de correo electrónico correctamente configurado, el comando `mail -s "Error de mantenimiento" root <<<"Error de tarea programada"` envía el mensaje de correo electrónico de aviso al usuario root. Tal comando podría ser usado en tareas desatendidas, como *cronjobs*, para informar al administrador del sistema sobre un problema inesperado. Escriba una construcción *if* que ejecutará comando mencionado `mail` si el estado de salida (sea cual sea) no tiene éxito.

Resumen

Esta lección cubre los conceptos básicos para comprender y escribir scripts de Bash shell. Los scripts de shell son una parte fundamental de cualquier distribución de Linux, ya que ofrecen una forma muy flexible de automatizar las tareas del usuario y del sistema que se realizan en el entorno de shell. En la lección se observaron los siguientes puntos:

- Estructura y permisos correctos en la creción de scripts shell
- Parámetros de script
- Usando variables para leer la entrada del usuario y para almacenar la salida de los comandos
- Arreglos en Bash
- Pruebas simples y ejecución condicional
- Formato de salida

Los comandos y procedimientos abordados fueron:

- Notación incorporada de Bash para sustitución de comandos, expansión de matrices y expresiones aritméticas
- Ejecución de comandos condicionales con los operadores `|| y `&&
- `echo`
- `chmod`
- `exec`
- `read`
- `declare`
- `test`
- `if`
- `printf`

Respuesta a los ejercicios guiados

1. La opción `-s` del comando `read` es útil para introducir contraseñas, ya que no mostrará el contenido que se está escribiendo en la pantalla. ¿Cómo podría usarse este comando para almacenar la entrada del usuario en la variable `PASSWORD` mientras se oculta el contenido escrito?

   `read -s PASSWORD`

2. El único propósito del comando `whoami` es mostrar el nombre de usuario que lo ha llamado, por lo que se utiliza principalmente dentro de los scripts para identificar al usuario que lo está ejecutando. Dentro de un script Bash, ¿cómo podría la salida del comando `whoami` ser almacenada en la variable llamada `WHO`?

   ``WHO=`whoami` `` or `WHO=$(whoami)`

3. ¿Qué operador de Bash debería estar entre los comandos `apt-get dist-upgrade` y `systemctl reboot`, si el usuario root quisiera ejecutar este último y solo si el comando `apt-get dist-upgrade` haya terminado con éxito?

   El operador `&&`, como en `apt-get dist-upgrade && systemctl reboot`.

Respuestas a los ejercicios de exploración

1. Después de intentar ejecutar un script Bash recién creado, el usuario recibe el siguiente mensaje de error:

   ```
   bash: ./script.sh: Permission denied
   ```

   Considerando que el archivo `./script.sh` fue creado por el mismo usuario, ¿cuál sería la causa probable de este error?

   El archivo `./script.sh` no tiene el permiso de ejecución habilitado.

2. Supongamos que un script llamado `do.sh` es ejecutable y el enlace simbólico llamado `do.sh` apunta a este. Dentro del script, ¿cómo podrías identificar si el nombre del archivo de llamada era `do.sh` o `undo.sh`?

   La variable especial `$0` contiene el nombre de archivo usado para llamar al script.

3. En un sistema con un servicio de correo electrónico correctamente configurado, el comando `mail -s "Error de mantenimiento" root <<<"Error de tarea programada"` envía el mensaje de correo electrónico de aviso al usuario root. Tal comando podría ser usado en tareas desatendidas, como *cronjobs*, para informar al administrador del sistema sobre un problema inesperado. Escriba una construcción *if* que ejecutará comando mencionado `mail` si el estado de salida (sea cual sea) no tiene éxito.

   `if [ "$?" -ne 0 ]; then mail -s "Maintenance Error" root <<<"Scheduled task error"; fi`



105.2 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	105 Shells y scripts
Objetivo:	105.2 Personalizar y escribir scripts sencillos
Lección:	2 de 2

Introducción

Las secuencias de comandos del Shell están generalmente destinadas a automatizar las operaciones relacionadas con los archivos y directorios, las mismas operaciones que podrían realizarse manualmente en la línea de comandos. Sin embargo, el alcance de los scripts de shell no sólo se limita a los documentos de un usuario, ya que la configuración e interacción con muchos aspectos de un sistema operativo Linux también se realiza a través de archivos de script.

El shell Bash ofrece comandos útiles para escribir scripts de shell, pero todo el poder de estos scripts depende de la combinación de los comandos incorporados de Bash con las diferentes utilidades de línea de comandos disponibles en un sistema Linux.

Pruebas ampliadas

El Bash como lenguaje de scripts está mayormente orientado a trabajar con archivos, por lo que el comando incorporado `test` tiene muchas opciones para evaluar las propiedades de los objetos del sistema de archivos (esencialmente archivos y directorios). Las pruebas que se centran en los archivos y directorios son útiles, por ejemplo, verificar si existen los archivos y directorios necesarios para realizar una determinada tarea y que los mismo se puedan leer. Luego, se asocia a

una construcción condicional *if* y se ejecuta un conjunto de acciones si la prueba tiene éxito.

El comando `test` puede evaluar expresiones usando dos sintaxis diferentes: las expresiones de prueba pueden darse como un argumento para el comando `test` o pueden colocarse entre corchetes, donde el comando `test` se da implícitamente. Así, la prueba para evaluar si `/etc` es un directorio válido puede escribirse como `test -d /etc` o como `[ -d /etc]`:

```
$ test -d /etc
$ echo $?
0
$ [ -d /etc ]
$ echo $?
0
```

Como confirman los códigos de salida, en la variable `$?`, un valor de 0 significa que la prueba fue exitosa, ambas formas evaluaron `/etc` como un directorio válido. Asumiendo que la ruta de un archivo o directorio fue almacenada en la variable `$VAR`, las siguientes expresiones pueden ser usadas como argumentos para `test` o dentro de los corchetes:

`-a "$VAR"`
: Evaluar si la ruta en `VAR` existe en el sistema de archivos y es un archivo.

`-b "$VAR"`
: Evaluar si la ruta en `VAR` es un archivo de bloque especial.

`-c "$VAR"`
: Evaluar si la ruta en `VAR` es un archivo de caracteres especiales.

`-d "$VAR"`
: Evaluar si la ruta en `VAR` es un directorio.

`-e "$VAR"`
: Evaluar si la ruta en `VAR` existe en el sistema de archivos.

`-f "$VAR"`
: Evaluar si la ruta en `VAR` existe y es un archivo regular.

`-g "$VAR"`
: Evaluar si la ruta en `VAR` tiene el permiso del SGID.

`-h "$VAR"`
: Evaluar si la ruta en `VAR` es un enlace simbólico.

`-L "$VAR"`
: Evaluar si la ruta en `VAR` es un enlace simbólico. (like `-h`).

`-k "$VAR"`
: Evaluar si la ruta en `VAR` tiene el permiso de *sticky*.

`-p "$VAR"`
: Evaluar si la ruta en `VAR` es un archivo *pipe*.

`-r "$VAR"`
: Evaluar si la ruta en `VAR` es legible por el usuario actual.

`-s "$VAR"`
: Evaluar si la ruta en `VAR` existe y no está vacía.

`-S "$VAR"`
: Evaluar si la ruta en `VAR` es un archivo socket.

`-t "$VAR"`
: Evaluar si la ruta en `VAR` está abierto en una terminal.

`-u "$VAR"`
: Evaluar si la ruta en `VAR` tiene el permiso SUID.

`-w "$VAR"`
: Evaluar si la ruta en `VAR` es escribible por el usuario actual.

`-x "$VAR"`
: Evaluar si la ruta en `VAR` es ejecutable por el usuario actual.

`-O "$VAR"`
: Evaluar si la ruta en `VAR` es propiedad del usuario actual.

`-G "$VAR"`
: Evaluar si la ruta en `VAR` pertenece al grupo del usuario actual.

`-N "$VAR"`
: Evaluar si la ruta en `VAR` ha sido modificado desde la última vez que se accedió.

`"$VAR1" -nt "$VAR2"`
: Evaluar si la ruta en `VAR1` es más nuevo que la ruta en el `VAR2`, según sus fechas de modificación.

`"$VAR1" -ot "$VAR2"`
: Evalúa si la ruta en el `VAR1` es más antiguo que el `VAR2`.

`"$VAR1" -ef "$VAR2"`
: Esta expresión evalúa a "True" si la ruta en `VAR1` es un enlace duro (hardlink) con `VAR2`.

Se recomienda usar las comillas dobles en una variable probada, porque si la variable resulta estar vacía, podría causar un error de sintaxis para el comando `test`. Las opciones de prueba requieren un argumento de operando, y una variable vacía sin comillas causaría un error debido a la falta de un argumento requerido. También hay pruebas para variables de texto arbitrarias, que se describen a continuación:

`-z "$TXT"`
: Evalúa si la variable `TXT` está vacía (tamaño cero).

`-n "$TXT"` o `test "$TXT"`
: Evalúa si la variable `TXT` no está vacía.

`"$TXT1" = "$TXT2"` or `"$TXT1" == "$TXT2"`
: Evalúa si la variable `TXT1` y `TXT2` son iguales.

`"$TXT1" != "$TXT2"`
: Evalúa si la variable `TXT1` y `TXT2` no son iguales..

`"$TXT1" < "$TXT2"`
: Evalúa si `TXT1` esta antes que `TXT2`, en orden alfabético.

`"$TXT1" > "$TXT2"`
: Evalúa si `TXT1` esta después que `TXT2`, en orden alfabético.

Los distintos idiomas pueden tener reglas diferentes para el orden alfabético. Para obtener resultados consistentes, independientemente de la configuración de localización del sistema donde se ejecuta el script, se recomienda establecer la variable de entorno `LANG` a `C`, ejemplo `LANG=C`, antes de realizar operaciones que impliquen un orden alfabético. Esta definición también mantendrá los mensajes del

sistema en el idioma original, por lo que debe ser usada sólo dentro del ámbito del script.

Las comparaciones numéricas tienen sus propias opciones de prueba:

`$NUM1 -lt $NUM2`
: Evalúa si `NUM1` es menor que `NUM2`.

`$NUM1 -gt $NUM2`
: Evalúa si `NUM1` es mayor que `NUM2`.

`$NUM1 -le $NUM2`
: Evalúa si `NUM1` es menor o igual que `NUM2`.

`$NUM1 -ge $NUM2`
: Evalúa si `NUM1` es mayor o igual que `NUM2`.

`$NUM1 -eq $NUM2`
: Evalúa si `NUM1` es igual a `NUM2`.

`$NUM1 -ne $NUM2`
: Evalúa si `NUM1` no es igual a `NUM2`.

Todas las pruebas pueden recibir los siguientes modificadores:

`! EXPR`
: Evalúa si la expresión `EXPR` es falsa.

`EXPR1 -a EXPR2`
: Evalúa si tanto `EXPR1` como `EXPR2` son verdaderos.

`EXPR1 -o EXPR2`
: Evalúa si al menos una de las dos expresiones es verdadera.

Otra construcción condicional es `case`, esta puede ser vista como una variación de *if*. La instrucción `case` ejecutará una lista de comandos dados si un ítem especificado,—ejemplo, el contenido de una variable—puede ser encontrado en una lista de ítems separados por *pipes* (la barra vertical `|`) y terminados por `)`. En el siguiente ejemplo, el script muestra cómo la construcción `case` puede ser usada para indicar el correspondiente formato de empaquetado para una distribución de Linux:

```
#!/bin/bash

DISTRO=$1

echo -n "Distribution $DISTRO uses "
case "$DISTRO" in
    debian | ubuntu | mint)
    echo -n "the DEB"
  ;;
    centos | fedora | opensuse )
    echo -n "the RPM"
  ;;
    *)
    echo -n "an unknown"
  ;;
esac
echo " package format."
```

Cada lista de patrones y los comandos asociados deben terminar con `;;`, `;&`, o `;;&`. El último patrón, un asterisco, coincidirá si ningún otro patrón anterior correspondió de antemano. La instrucción `esac` (*case* al revés) termina la construcción `case`. Asumiendo que el script anterior se llamaba `script.sh` y se ejecuta con `opensuse` como primer argumento, se generará la siguiente salida:

```
$ ./script.sh opensuse
Distribution opensuse uses the RPM package format.
```

TIP Bash tiene una opción llamada `nocasematch` que permite la coincidencia de patrones no sensibles a mayúsculas y minúsculas para la construcción de `case` y otros comandos condicionales. El comando incorporado `shopt` cambia los valores de las configuraciones que controlan el comportamiento opcional del shell: `shopt -s` habilitará (*set*) la opción dada y `shopt -u` deshabilitará (*unset*) la opción dada. Por lo tanto, si se coloca `shopt -s nocasematch` antes de la construcción de mayúsculas y minúsculas, se habilitará la coincidencia de patrones no sensibles a estas. Las opciones modificadas por `shopt` sólo afectarán a la sesión actual, por lo que las opciones modificadas dentro de los scripts que se ejecutan en una sub-shell — que es la forma estándar de ejecutar un script — no afecta las opciones de la sesión padre.

El elemento buscado y los patrones se someten a la expansión de la tilde, la expansión de los parámetros, la sustitución de los comandos y la expansión aritmética. Si el elemento buscado se especifica con comillas, se eliminarán antes de que se intente la coincidencia.

Construcciones de bucle

Los scripts se utilizan a menudo como herramienta para automatizar tareas repetitivas, realizando el mismo conjunto de comandos hasta que se verifique un criterio. Bash tiene tres instrucciones de bucle — `for`, `until` y `while` — diseñadas para construcciones de bucle ligeramente distintas.

La construcción `for` camina a través de una lista dada de elementos — usualmente una lista de palabras o cualquier otro segmento de texto separado del espacio — ejecutando el mismo conjunto de comandos en cada uno de esos elementos. Antes de cada iteración, la instrucción `for` asigna el elemento actual a una variable, que puede ser utilizada por los comandos incluidos. El proceso se repite hasta que no quedan más ítems. La sintaxis de la construcción `for` es:

```
for VARNAME in LIST
do
    COMMANDS
done
```

`VARNAME` es un nombre arbitrario de una variable de shell y `LIST` es cualquier secuencia de términos separados. Los caracteres delimitadores válidos que dividen los elementos de la lista están definidos por la variable de entorno `IFS`, que son los caracteres *espacio*, *tabulación* y *nueva línea* por defecto. La lista de comandos a ejecutar está delimitada por las instrucciones `do` y `done`, por lo que los comandos pueden ocupar tantas líneas como sean necesarias.

En el siguiente ejemplo, el comando `for` tomará cada elemento de la lista proporcionada — una secuencia de números — y lo asignará a la variable `NUM`, un elemento a la vez:

```
#!/bin/bash

for NUM in 1 1 2 3 5 8 13
do
    echo -n "$NUM is "
    if [ $(( $NUM % 2 )) -ne 0 ]
    then
        echo "odd."
    else
        echo "even."
  fi
done
```

En el ejemplo, un constructo anidado `if` se utiliza junto con una expresión aritmética para evaluar si el número de la variable actual `NUM` es par o impar. Asumiendo que el anterior script de muestra se

llamaba `script.sh` y está en el directorio actual, se generará la siguiente salida:

```
$ ./script.sh
1 is odd.
1 is odd.
2 is even.
3 is odd.
5 is odd.
8 is even.
13 is odd.
```

Bash también apoya un formato alternativo a las construcciones `for`, con la notación de doble paréntesis. Esta notación se asemeja a la sintaxis de la instrucción `for` del lenguaje de programación C y es particularmente útil para trabajar con arreglos:

```
#!/bin/bash

SEQ=( 1 1 2 3 5 8 13 )

for (( IDX = 0; IDX < ${#SEQ[*]}; IDX++ ))
do
    echo -n "${SEQ[$IDX]} is "
    if [ $(( ${SEQ[$IDX]} % 2 )) -ne 0 ]
    then
        echo "odd."
    else
        echo "even."
  fi
done
```

Este script de muestra, generará exactamente la misma salida que el ejemplo anterior. Sin embargo, en lugar de usar la variable `NUM` para almacenar un elemento a la vez, la variable `IDX` se emplea para rastrear el índice de la matriz actual en orden ascendente, comenzando desde 0 y añadiéndole continuamente mientras está bajo el número de elementos de la matriz `SEQ`. El ítem actual se recupera de su posición en la matriz con `${SEQ[$IDX]}`.

De la misma manera, la construcción `until` ejecuta una secuencia de comandos hasta que un comando de prueba — como el propio comando `test` — termina con el estado 0 (éxito). Por ejemplo, la misma estructura de bucle del ejemplo anterior puede implementarse con `until`:

```
#!/bin/bash

SEQ=( 1 1 2 3 5 8 13 )

IDX=0

until [ $IDX -eq ${#SEQ[*]} ]
do
    echo -n "${SEQ[$IDX]} is "
    if [ $(( ${SEQ[$IDX]} % 2 )) -ne 0 ]
    then
        echo "odd."
    else
        echo "even."
  fi
  IDX=$(( $IDX + 1 ))
done
```

Las construcciones `until` pueden requerir más instrucciones que las de `for`, pero puede ser más adecuado para los criterios (a la hora detener el bucle) no numéricos proporcionados por las expresiones de `test` o cualquier otro comando. Es importante incluir acciones que aseguren un criterio de parada válido, como el incremento de una variable de contador, ya que de lo contrario el bucle puede ejecutarse indefinidamente.

La instrucción `while` es similar a la instrucción `until`, pero `while` sigue repitiendo el conjunto de comandos si el comando de prueba termina con el estado 0 (éxito). Por lo tanto, la instrucción until [$IDX -eq ${#SEQ[*]}]` del ejemplo anterior es equivalente a `while [$IDX -lt ${#SEQ[*]}]`, ya que el bucle debe repetirse mientras el índice de la matriz es menor que el total de los elementos de esta.

Un ejemplo más elaborado

Imagine que un usuario quiere sincronizar periódicamente una colección de sus archivos y directorios con otro dispositivo de almacenamiento montado en el sistema de archivos; dado que un sistema de respaldo con todas las funciones se considera una exageración. Dado que esta es una actividad que debe realizarse periódicamente, es una buena aplicación candidata para automatizar con un script de shell.

La tarea es sencilla: sincronizar cada archivo y directorio contenido en una lista, desde un directorio de origen informado como primer argumento en el script hasta un directorio de destino informado como segundo argumento de este. Para facilitar la adición o eliminación de elementos de la lista, se mantendrá en un archivo separado; un elemento por línea:

```
$ cat ~/.sync.list
Documents
To do
Work
Family Album
.config
.ssh
.bash_profile
.vimrc
```

El archivo contiene una mezcla de archivos y directorios, algunos con espacios en blanco en sus nombres. Este es un escenario adecuado para el comando incorporado de Bash `mapfile`, que analizará cualquier contenido de texto y creará una variable de matriz a partir de este, colocando cada línea como un elemento de matriz individual. El archivo de script se llamará `synnc.sh`, conteniendo el siguiente contenido:


```
#!/bin/bash

set -ef

# List of items to sync
FILE=~/.sync.list

# Origin directory
FROM=$1

# Destination directory
TO=$2

# Check if both directories are valid
if [ ! -d "$FROM" -o ! -d "$TO" ]
then
  echo Usage:
  echo "$0 <SOURCEDIR> <DESTDIR>"
  exit 1
fi

# Create array from file
mapfile -t LIST < $FILE

# Sync items
for (( IDX = 0; IDX < ${#LIST[*]}; IDX++ ))
do
    echo -e "$FROM/${LIST[$IDX]} \u2192 $TO/${LIST[$IDX]}";
    rsync -qa --delete "$FROM/${LIST[$IDX]}" "$TO";
done
```

La primera acción que hace el script es redefinir dos parámetros de shell con el comando `set`: la opción `-e` saldrá de la ejecución inmediatamente si un comando sale con un estado distinto de cero y la opción `-f` deshabilitará el globbing de nombres de archivo. Ambas opciones se pueden acortar con `-ef`. Este no es un paso obligatorio, pero ayuda a disminuir la probabilidad de un comportamiento inesperado.

Las instrucciones reales orientadas a la aplicación del archivo de script pueden dividirse en tres partes:

Recolectar y comprobar los parámetros del script

+ La variable `FILE` es la ruta del archivo que contiene la lista de elementos a copiar: `~/.sync.list`.

Las variables `FROM` y `TO` son el origen y el destino, respectivamente. Dado que estos dos últimos parámetros son proporcionados por el usuario, pasan por una simple prueba de validación realizada por la construcción `if`: si alguno de los dos no es un directorio válido — evaluado por la prueba [! -d "$FROM" -o ! -d "$TO"]` — el script mostrará un breve mensaje de ayuda y luego terminará con un estado de salida de 1.

1. *Carga la lista de archivos y directorios*

   Después de definir todos los parámetros, se crea un arreglo que contiene la lista de elementos a copiar con el comando `mapfile -t LIST < $FILE`. La opción `-t` de `mapfile` eliminará el carácter de la nueva línea (newline) de cada línea antes de incluirla en la variable del arreglo `LIST`. El contenido del archivo indicado por la variable `FILE` — `~/.sync.list` — se lee a través de la redirección de entrada.

2. *Realizar la copia e informar al usuario*

   Un bucle `for` usando notación de doble paréntesis atraviesa el conjunto de elementos, con la variable `IDX` llevando la cuenta incremental del índice. El comando `echo` informará al usuario de cada elemento que se está copiando. El carácter unicode escape — `\u2192` — para el carácter *right arrow* está presente en el mensaje de salida, por lo que la opción `-e` del comando `echo` debe ser usada. El comando `rsync` copiará selectivamente sólo las piezas del archivo modificado desde el origen, por lo que se recomienda su uso para tales tareas. Las opciones de `rsync` como `q` y `a`, condensadas en `qa`, inhibirán los mensajes `rsync` y activarán el modo `archivo`, donde se conservan todas las propiedades del archivo. La opción `--delete` hará que `rsync` elimine un elemento en el destino que ya no existe en el origen, por lo que debe ser usado con cuidado.

Asumiendo que todos los elementos de la lista existen en el directorio principal del usuario `carol`, `/home/carol`, y que el directorio de destino `/media/carol/backup` apunta a un dispositivo de almacenamiento externo montado, el comando `sync.sh /home/carol /media/carol/backup` generará la siguiente salida:

```
$ sync.sh /home/carol /media/carol/backup
/home/carol/Documents → /media/carol/backup/Documents
/home/carol/"To do" → /media/carol/backup/"To do"
/home/carol/Work → /media/carol/backup/Work
/home/carol/"Family Album" → /media/carol/backup/"Family Album"
/home/carol/.config → /media/carol/backup/.config
/home/carol/.ssh → /media/carol/backup/.ssh
/home/carol/.bash_profile → /media/carol/backup/.bash_profile
/home/carol/.vimrc → /media/carol/backup/.vimrc
```

El ejemplo también supone que el guión se ejecuta por root o por el usuario `carol`, ya que la mayoría de los archivos serían ilegibles para otros usuarios. Si `script.sh` no está dentro de un directorio listado en la variable de entorno `PATH`, entonces debe ser especificado con su ruta completa.

Ejercicios guiados

1. ¿Cómo podría usarse el comando `test` para verificar si la ruta del archivo almacenado en la variable `FROM` es más reciente que el archivo cuya ruta está en la variable `TO`?

2. El siguiente guión debe imprimir una secuencia numérica del 0 al 9, pero en cambio imprime indefinidamente el 0. ¿Qué se debe hacer para obtener el resultado esperado?

```
#!/bin/bash

COUNTER=0

while [ $COUNTER -lt 10 ]
do
  echo $COUNTER
done
```

3. Supongamos que un usuario escribió un script que requiere una lista ordenada de nombres de usuario. La lista ordenada resultante se presenta como la siguiente en su computadora:

```
carol
Dave
emma
Frank
Grace
henry
```

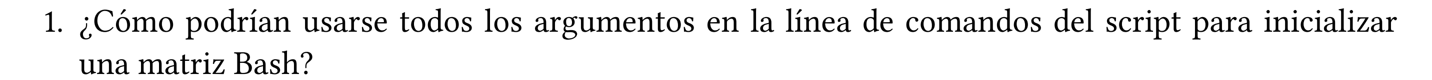
Sin embargo, la misma lista está ordenada como la siguiente en la computadora de su colega:

```
Dave
Frank
Grace
carol
emma
henry
```

¿Cómo podría explicar las diferencias entre las dos listas clasificadas?

Ejercicios de exploración

1. ¿Cómo podrían usarse todos los argumentos en la línea de comandos del script para inicializar una matriz Bash?



```
¿Por qué, contrariamente a la intuición, el comando `test 1 > 2` se evalúa como
verdadero?
```

2. ¿Cómo podría un usuario cambiar temporalmente el separador de campos predeterminado por el carácter de nueva línea solamente, sin dejar de poder revertirlo a su contenido original?

Resumen

Esta lección profundiza en las pruebas disponibles para el comando `test` y en otras construcciones condicionales y de bucle necesarias para escribir scripts de shell más elaborados. Se da un simple script de sincronización de archivos como ejemplo de una aplicación práctica de shell script. La lección abarcó los siguientes pasos:

- Pruebas extendidas para las construcciones condicionales `if` y `case`.
- Construcciones de bucle en shell: `for`, `until` y `while`.
- Iterando a través de arreglos y parámetros.

Los comandos y procedimientos abordados fueron:

`test`
: Realiza una comparación entre los artículos suministrados al comando.

`if`
: Una construcción lógica utilizada en scripts para evaluar algo como verdadero o falso, luego bifurca la ejecución del comando en función de los resultados.

`case`
: Evalúa varios valores frente a una sola variable. La ejecución del comando de secuencia de comandos se lleva a cabo dependiendo del resultado del comando `case`.

`for`
: Repite la ejecución de un comando según un criterio dado.

`until`
: Repite la ejecución de un comando hasta que una expresión se evalúe como falsa.

`while`
: Repite la ejecución de un comando mientras una expresión dada se evalúa como verdadera.

Respuesta a los ejercicios guiados

1. ¿Cómo podría usarse el comando `test` para verificar si la ruta del archivo almacenado en la variable `FROM` es más reciente que el archivo cuya ruta está en la variable `TO`?

   El comando `test "$ FROM" -nt "$TO"` devolverá un código de estado en 0 si el archivo en la variable `FROM` es más reciente que el archivo en la variable `TO`.

2. El siguiente guión debe imprimir una secuencia numérica del 0 al 9, pero en cambio imprime indefinidamente el 0. ¿Qué se debe hacer para obtener el resultado esperado?

   ```
   #!/bin/bash

   COUNTER=0

   while [ $COUNTER -lt 10 ]
   do
     echo $COUNTER
   done
   ```

   La variable `COUNTER` debe ser incrementada, lo que podría hacerse con la expresión aritmética `COUNTER=$(( $COUNTER + 1 ))`, para eventualmente alcanzar el criterio de parada y terminar el bucle.

3. Supongamos que un usuario escribió un script que requiere una lista ordenada de nombres de usuario. La lista ordenada resultante se presenta como la siguiente en su computadora:

   ```
   carol
   Dave
   emma
   Frank
   Grace
   henry
   ```

   Sin embargo, la misma lista está ordenada como la siguiente en la computadora de su colega:



```
Dave
Frank
Grace
carol
emma
henry
```

¿Cómo podría explicar las diferencias entre las dos listas clasificadas?

La clasificación se basa en la ubicación del sistema actual. Para evitar las inconsistencias, las tareas de clasificación deben ser realizadas con la variable de entorno `LANG` puesta en `C`.

Respuestas a los ejercicios de exploración

1. ¿Cómo podrían usarse todos los argumentos en la línea de comandos del script para inicializar una matriz Bash?

   Los comandos `PARAMS=( $* )` o `PARAMS=( "$@" )` crearán una matriz llamada `PARAMS` con todos los argumentos.

2. ¿Por qué, contrariamente a la intuición, el comando `test 1 > 2` se evalúa como verdadero?

   El operador `>` está pensado para ser usado con cadenas de caracteres, no con pruebas numéricas.

3. ¿Cómo podría un usuario cambiar temporalmente el separador de campos predeterminado por el carácter de nueva línea solamente, sin dejar de poder revertirlo a su contenido original?

   Una copia de la variable `IFS` puede ser almacenada en otra variable: `OLDIFS=$IFS`. Entonces el nuevo separador de línea se define con `IFS=$'\n'` y la variable IFS puede ser revertida con `IFS=$OLDIFS`.



Tema 106: Interfaces de usuario y escritorios



106.1 Instalar y configurar X11

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 106.1

Importancia

2

Áreas de conocimiento clave

- Entender la arquitectura de X11.
- Conocimientos básicos del archivo de configuración de X Window.
- Sobreescribir aspectos específicos de la configuración de Xorg, tales como la configuración del teclado.
- Entender los componentes de los entornos de escritorio, tales como los gestores de pantalla o los gestores de ventanas.
- Gestionar el acceso al servidor X y a las aplicaciones de pantalla en servidores X remotos.

Lista parcial de archivos, términos y utilidades

- `/etc/X11/xorg.conf`
- `/etc/X11/xorg.conf.d/`
- `~/.xsession-errors`
- `xhost`
- `xauth`
- `DISPLAY`
- `X`



106.1 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	106 Interfaces de usuario y escritorios
Objetivo:	106.1 Instalar y configurar X11
Lección:	1 de 1

Introducción

El Sistema X Window es una pila de software que se utiliza para mostrar texto y gráficos en una pantalla. El aspecto y diseño de un cliente X no está dictado por el Sistema X Window, sino que es manejado por cada cliente X individual, un *administrador de ventanas* (por ejemplo, Window Maker, Tab Window Manager), o un *entorno de escritorio* completo como KDE, GNOME, o Xfce. En otra lección trataremos los entornos de escritorio. Lo que ahora haremos es centrarnos en la arquitectura subyacente y en las herramientas comunes para el sistema X Window que un administrador usaría para configurar X.

El sistema X Window es multiplataforma y funciona en varios sistemas operativos como Linux, BSD, Solaris y otros sistemas de tipo Unix. También hay implementaciones disponibles para el MacOS de Apple y Windows de Microsoft.

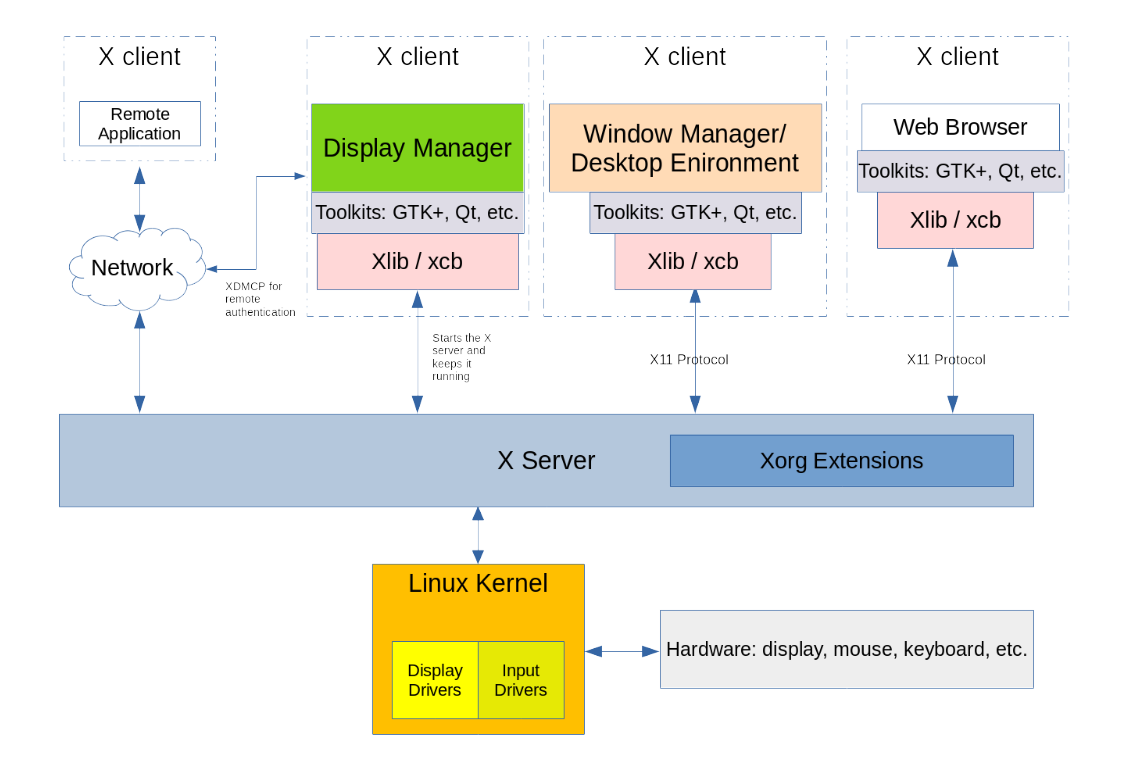
La versión primaria del protocolo X usado en las distribuciones modernas de Linux es la versión 11 de *X.org*, comúnmente escrita como *X11*. El protocolo X es el mecanismo de comunicación entre el cliente X y el servidor X. Las diferencias entre ellos se discutirán más adelante.

> **NOTE**
>
> El predecesor del Sistema X Window fue un sistema de ventanas llamado *W* y fue un esfuerzo en conjunto entre IBM, DEC y MIT. Este software nació del Proyecto Athena en 1984. Cuando los desarrolladores empezaron a trabajar en un nuevo servidor de ventanas, eligieron la siguiente letra del alfabeto inglés: “X”. La evolución del Sistema X WIndow está actualmente controlada por el *MIT X Consortium*.

Arquitectura de sistema X Window

El Sistema X Window proporciona los mecanismos para dibujar formas bidimensionales básicas (y tridimensionales a través de extensiones) en una pantalla. Se divide en un cliente y un servidor; en la mayoría de las instalaciones en las que se requiere un escritorio gráfico, estos componentes estarán presentes en el mismo equipo. El componente cliente toma la forma de una aplicación, como un emulador de terminal, un juego o un navegador web. Cada aplicación cliente informa al servidor X sobre la ubicación y el tamaño de su ventana en la pantalla de una computadora. El cliente también maneja lo que entra en esa ventana, y el servidor X coloca el dibujo solicitado en la pantalla. El sistema X WIndow también maneja la entrada de dispositivos como mouse, teclados, trackpads y más.

Estructura básica de un sistema X Window



El sistema X Window es capaz de funcionar en red donde múltiples clientes X de diferentes computadoras de una red pueden hacer solicitudes a un solo servidor X remoto. El razonamiento que subyace a esto es que un administrador o usuario puede tener acceso a una aplicación gráfica en un sistema remoto que puede no estar disponible en su sistema local. X Window es un sistema modular, y esto es una característica clave. A lo largo de la existencia de este sistema se han desarrollado nuevas características que se han añadido a su marco (Framework). Estos nuevos componentes sólo fueron añadidos como extensiones al servidor X, dejando el núcleo del protocolo X11 intacto. Estas extensiones están contenidas dentro de los archivos de la biblioteca *Xorg*. Algunos ejemplos de bibliotecas Xorg incluyen: `libXrandr`, `libXcursor`, `libX11`, `libxkbfile` así como otras, cada una de ellas proporcionan una funcionalidad extendida al servidor X.

Un *administrador de pantalla* proporciona un acceso gráfico a un sistema. Este sistema puede ser un ordenador local o un ordenador a través de una red. El administrador de pantalla se lanza después de que la computadora se inicia y así comenzará una sesión de servidor X para el usuario autenticado. El administrador de pantalla también es responsable de mantener el servidor X en funcionamiento. Algunos ejemplos de gestores de pantalla son: GDM, SDDM y LightDM. Cada instancia de un servidor X en funcionamiento tiene un *nombre de pantalla* para identificarlo. El nombre de pantalla contiene lo siguiente:

```
hostname:displaynumber.screennumber
```

El nombre de la pantalla también indica una aplicación gráfica donde debe ser renderizada y sobre cúal host (si se utiliza una conexión X remota).

El `hostname` se refiere al nombre del sistema que mostrará la aplicación. Si falta un nombre de host en el nombre de pantalla, entonces se asume que es el host local.

El `displaynumber` hace referencia a la colección de “pantallas” que están en uso, ya sea una sola o varias pantallas en una estación de trabajo. A cada sesión de servidor X en ejecución se le da un número de pantalla que comienza en `0`.

Por defecto el `screennumber` es `0`. Esto puede ser el caso si sólo una pantalla o varias pantallas físicas están configuradas para trabajar como una sola pantalla. Cuando todas las pantallas en una configuración de múltiples monitores se combinan en una lógica, las ventanas de aplicación pueden moverse libremente entre las pantallas. En situaciones en las que cada pantalla está configurada para funcionar de forma independiente, cada una albergará las ventanas de aplicación que se abran dentro de ella y las ventanas no podrán moverse de una pantalla a otra. Cada pantalla independiente tendrá su propio número asignado. Si sólo hay una lógica en uso, entonces se omite el punto y el número de pantalla.

El nombre de una sesión X en curso se almacena en la variable de entorno `DISPLAY`:

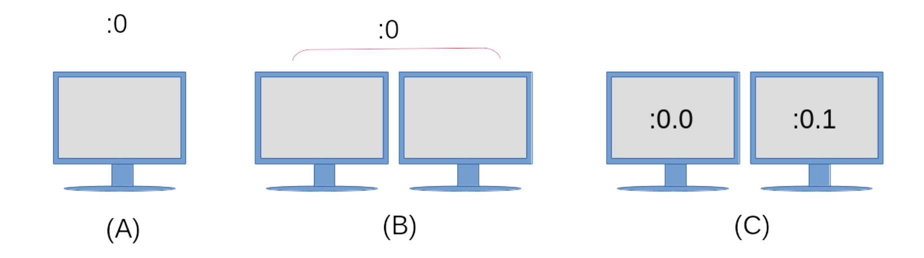
```
$ echo $DISPLAY
:0
```

La salida detalla lo siguiente:

1. El servidor X en uso está en el sistema local, por lo tanto no hay nada impreso a la izquierda de los dos puntos.
2. La sesión actual del servidor X es la primera como indica el `0` que sigue a los dos puntos.
3. Sólo hay una pantalla lógica en uso, por lo que un número de pantalla no es visible.

Para ilustrar mejor este concepto, véase el siguiente diagrama:

Ejemplo de configuración en pantalla



(A)
Un solo monitor, con una sola configuración de visualización y una sola pantalla.

(B)
Configurado como una sola pantalla, con dos monitores físicos como una sola. Las ventanas de la aplicación pueden moverse libremente entre los dos monitores.

(C)
Una configuración de pantalla única (como se indica en `:0`) sin embargo, cada monitor es una pantalla independiente. Ambas pantallas seguirán compartiendo los mismos dispositivos de entrada como el teclado y el mouse, pero una aplicación abierta en la pantalla `:0.0` no puede ser movida a la pantalla `:0.1` o viceversa.

Para iniciar una aplicación en una pantalla específica, asigne el número de pantalla a la variable de entorno `DISPLAY` antes de lanzar la aplicación:

```
$ DISPLAY=:0.1 firefox &
```

Este comando iniciaría el navegador web Firefox en la pantalla de la derecha, en el diagrama de arriba. Algunos conjuntos de herramientas también proporcionan opciones de línea de comandos para ordenar a una aplicación que se ejecute en una pantalla especifica. Revise un ejemplo de `--screen` y `--display` en la página de manual de `gtk-options(7)`.

Configuración de un servidor X

Tradicionalmente, el principal archivo de configuración que se utiliza para configurar un servidor X es el archivo `/etc/X11/xorg.conf`. En las distribuciones modernas de Linux, el servidor X se configurará así mismo en tiempo de ejecución cuando este es iniciado, y por lo tanto no puede existir ningún archivo `xorg.conf`.

El archivo `xorg.conf` está dividido en estrofas llamadas `secciones`. Cada sección comienza con el término `Section` y después de este término se encuentra el *nombre de la sección* que se refiere a la configuración de un componente. Cada `Section` está correspondientemente terminada por una `EndSection`. El típico archivo `xorg.conf` contiene las siguientes secciones:

`InputDevice`
Se utiliza para configurar un modelo específico de teclado o mouse.

`InputClass`
En las distribuciones modernas de Linux esta sección se encuentra típicamente en un archivo de configuración separado y localizado en `/etc/X11/xorg.conf.d/`. El `InputClass` se usa para configurar una *clase* de dispositivos de hardware como teclados y mouses en lugar de un componente específico de hardware. Abajo encontraremos un ejemplo del archivo `/etc/X11/xorg.conf.d/00-keyboard.conf`:

```
Section "InputClass"
        Identifier "system-keyboard"
        MatchIsKeyboard "on"
        Option "XkbLayout" "us"
        Option "XkbModel" "pc105"
EndSection
```

La opción para `XkbLayout` determina la disposición de las teclas de un teclado, como Dvorak, para diestros y zurdos, QWERTY e idioma. La opción para `XkbModel` se usa para definir el tipo de teclado en uso. Una tabla de modelos, diseños y sus descripciones se puede encontrar en

`xkeyboard-config(7)`. Los archivos asociados a las distribuciones de teclado se pueden encontrar en `/usr/share/X11/xkb`. Un ejemplo de diseño de teclado griego politónico en una computadora de Chromebook se vería como el siguiente:

```
Section "InputClass"
        Identifier "system-keyboard"
        MatchIsKeyboard "on"
        Option "XkbLayout" "gr(polytonic)"
        Option "XkbModel" "chromebook"
EndSection
```

Alternativamente, la disposición del teclado puede ser modificada durante una sesión X en curso con el comando `setxkbmap`. Aquí hay un ejemplo de este comando que configura la disposición del Politónico Griego en una computadora Chromebook:

```
$ setxkbmap -model chromebook -layout "gr(polytonic)"
```

Este ajuste sólo durará mientras la sesión X esté en uso. Para que estos cambios sean permanentes, modifique el archivo `/etc/X11/xorg.conf.d/00-keyboard.conf` para incluir los ajustes necesarios.

NOTE El comando `setxkbmap` utiliza *X Keyboard Extension* (XKB). Este es un ejemplo de la funcionalidad aditiva del Sistema X Window a través de su uso de extensiones.

Las distribuciones modernas de Linux proporcionan el comando `localectl` a través de `systemd` que también puede ser usado para modificar una disposición del teclado y creará automáticamente el archivo de configuración `/etc/X11/xorg.conf.d/00-keyboard.conf`. Una vez más, aquí hay un ejemplo de configuración de un teclado politónico griego en un Chromebook, esta vez usando el comando `localectl`:

```
$ localectl --no-convert set-x11-keymap "gr(polytonic)" chromebook
```

La opción `--no-convert` se usa aquí para evitar que `localectl` modifique el mapa de teclas de la consola del host.

`Monitor`

La sección `Monitor` describe el monitor físico que se utiliza y dónde está conectado. El siguiente

es un ejemplo de configuración que muestra un monitor físico conectado al segundo puerto de pantalla y que se utiliza como monitor primario.

```
Section "Monitor"
        Identifier  "DP2"
        Option      "Primary" "true"
EndSection
```

`Device`

La sección `Device` describe la tarjeta de vídeo física que se utiliza. También contendrá el módulo del núcleo utilizado como controlador de la tarjeta de vídeo, junto con su ubicación física en la placa base.

```
Section "Device"
        Identifier  "Device0"
        Driver      "i915"
        BusID       "PCI:0:2:0"
EndSection
```

`Screen`

La sección `Screen` une las secciones `Monitor` y `Device`. Un ejemplo de la sección `Screen` podría ser la siguiente;

```
Section "Screen"
        Identifier "Screen0"
        Device     "Device0"
        Monitor    "DP2"
EndSection
```

`ServerLayout`

La sección `ServerLayout` agrupa todas las secciones como el mouse, el teclado y las pantallas en una interfaz del sistema X Window.

```
Section "ServerLayout"
    Identifier   "Layout-1"
    Screen       "Screen0" 0 0
    InputDevice  "mouse1"  "CorePointer"
    InputDevice  "system-keyboard"  "CoreKeyboard"
EndSection
```

NOTE No todas las secciones pueden encontrarse dentro de un archivo de configuración. En los casos en que falta una sección, los valores por defecto los proporciona la instancia del servidor X que se está ejecutando.

Los archivos de configuración especificados por el usuario también residen en `/etc/X11/xorg.conf.d/`. Los archivos de configuración proporcionados por la distribución se localizan en `/usr/share/X11/xorg.conf.d/`. Los archivos de configuración ubicados dentro de `/etc/X11/xorg.conf.d/` son analizados antes del archivo `/etc/X11/xorg.conf` si existe en el sistema.

El comando `xdpyinfo` se usa en una computadora para mostrar información sobre una instancia de servidor X en ejecución. A continuación se muestra un ejemplo de la salida del comando:

```
$ xdpyinfo
name of display:    :0
version number:    11.0
vendor string:    The X.Org Foundation
vendor release number:    12004000
X.Org version: 1.20.4
maximum request size:  16777212 bytes
motion buffer size:  256
bitmap unit, bit order, padding:    32, LSBFirst, 32
image byte order:    LSBFirst
number of supported pixmap formats:    7
supported pixmap formats:
    depth 1, bits_per_pixel 1, scanline_pad 32
    depth 4, bits_per_pixel 8, scanline_pad 32
    depth 8, bits_per_pixel 8, scanline_pad 32
    depth 15, bits_per_pixel 16, scanline_pad 32
    depth 16, bits_per_pixel 16, scanline_pad 32
    depth 24, bits_per_pixel 32, scanline_pad 32
    depth 32, bits_per_pixel 32, scanline_pad 32
keycode range:    minimum 8, maximum 255
focus:  None
number of extensions:    25
    BIG-REQUESTS
```

```
    Composite
    DAMAGE
    DOUBLE-BUFFER
    DRI3
    GLX
    Generic Event Extension
    MIT-SCREEN-SAVER
    MIT-SHM
    Present
    RANDR
    RECORD
    RENDER
    SECURITY
    SHAPE
    SYNC
    X-Resource
    XC-MISC
    XFIXES
    XFree86-VidModeExtension
    XINERAMA
    XInputExtension
    XKEYBOARD
    XTEST
    XVideo
default screen number:    0
number of screens:    1

screen #0:
  dimensions:    3840x1080 pixels (1016x286 millimeters)
  resolution:    96x96 dots per inch
  depths (7):    24, 1, 4, 8, 15, 16, 32
  root window id:    0x39e
  depth of root window:    24 planes
  number of colormaps:    minimum 1, maximum 1
  default colormap:    0x25
  default number of colormap cells:    256
  preallocated pixels:    black 0, white 16777215
  options:    backing-store WHEN MAPPED, save-unders NO
  largest cursor:    3840x1080
  current input event mask:    0xda0033
    KeyPressMask             KeyReleaseMask           EnterWindowMask
    LeaveWindowMask          StructureNotifyMask      SubstructureNotifyMask
    SubstructureRedirectMask PropertyChangeMask       ColormapChangeMask
  number of visuals:    270
...
```

Las partes más relevantes de la salida están en negrita, como el nombre de la pantalla (que es el mismo que el contenido de la variable de entorno `DISPLAY`), la versión del servidor X en uso, el número y listado de las extensiones Xorg en uso, y más información sobre la propia pantalla.

Creando un archivo de configuración básica de Xorg

Aunque X creará su configuración después del inicio del sistema en instalaciones modernas de Linux, un archivo `xorg.conf` todavía puede ser usado. Para generar un archivo `/etc/X11/xorg.conf` permanente, ejecute el siguiente comando:

```
$ sudo Xorg -configure
```

NOTE

Si ya existe una sesión X en ejecución, tendrás que especificar un `DISPLAY` diferente en tu comando, por ejemplo:

```
$ sudo Xorg :1 -configure
```

En algunas distribuciones de Linux, el comando `X` puede ser usado en lugar de `Xorg`, ya que `X` es un enlace simbólico a `Xorg`.

Se creará un archivo `xorg.conf.new` en su actual directorio de trabajo. El contenido de este archivo se deriva de lo que el servidor X ha encontrado disponible en el hardware y los controladores del sistema local. Para usar este archivo, tendrá que ser movido al directorio `/etc/X11/` y renombrado a `xorg.conf`:

```
$ sudo mv xorg.conf.new /etc/X11/xorg.conf
```

NOTE

Las siguientes páginas del manual proporcionan más información sobre los componentes del Sistema X Window: `xorg.conf(5)`, `Xserver(1)`, `X(1)` and `Xorg(1)`.

Wayland

Wayland es el nuevo protocolo de visualización diseñado para reemplazar el Sistema X Window. Muchas distribuciones modernas de Linux lo usan como su servidor de visualización por defecto. Se supone que es más ligero en cuanto a recursos del sistema y su instalación ocupa menos espacio en disco que X. El proyecto comenzó en 2010 y todavía está en desarrollo activo, incluyendo el trabajo de los desarrolladores activos y anteriores de X.org.

A diferencia del sistema X Window, no hay ninguna instancia de servidor que se ejecute entre el cliente y el kernel. En su lugar, una ventana cliente trabaja con su propio código o el de un kit de herramientas (como Gtk+ o Qt) para proporcionar el renderizado. Para hacer el renderizado, se hace una petición al kernel de Linux a través del protocolo Wayland. El kernel reenvía la solicitud a través de este protocolo al Wayland *compositor* que se encarga de la entrada de dispositivos, la gestión de ventanas y la composición. El compositor es la parte del sistema que combina los elementos renderizados en una salida visual en la pantalla.

La mayoría de las herramientas modernas como Gtk+ 3 y Qt 5 han sido actualizadas para permitir la renderización a un sistema X Window o a una computadora con Wayland. No todas las aplicaciones autónomas han sido escritas para soportar el renderizado en Wayland hasta ahora. Para las aplicaciones y frameworks que todavía tienen como objetivo que se ejecute el Sistema X Window, la aplicación puede ejecutarse dentro de *XWayland*. El sistema XWayland es un servidor X separado que se ejecuta dentro de un cliente de Wayland y por lo tanto, renderiza el contenido de una ventana de cliente dentro de una instancia de servidor X independiente.

Así como el sistema X Window utiliza una variable de entorno `DISPLAY` para controlar las pantallas en uso, el protocolo Wayland utiliza una variable de entorno `WAYLAND_DISPLAY`. A continuación se muestra la salida de un sistema que ejecuta una pantalla Wayland:

```
$ echo $WAYLAND_DISPLAY
wayland-0
```

Esta variable de entorno no está disponible en los sistemas que ejecutan X.

Guided Exercises

1. ¿Qué comando utilizaría para determinar qué extensiones Xorg están disponibles en un sistema?

2. Acaba de recibir un nuevo mouse de 10 botones para su computadora, sin embargo, requerirá una configuración extra para que todos los botones funcionen correctamente. Sin modificar el resto de la configuración del servidor X. ¿Qué directorio utilizaría para crear un nuevo archivo de configuración para este mouse y qué sección de configuración específica se utilizaría en este archivo?

3. ¿Qué componente de una instalación de Linux es responsable de mantener un servidor X funcionando?

4. ¿Qué opción en la línea de comandos se utiliza con el comando `X` para crear un nuevo archivo de configuración `xorg.conf`?

Explorational Exercises

1. ¿Cuál sería el contenido de la variable de entorno `DISPLAY` en un sistema llamado `lab01` con una configuración de pantalla única? Supongamos que la variable de entorno `DISPLAY` se visualiza en un emulador de terminal en la tercera pantalla independiente.

2. ¿Qué comando se puede usar para crear un archivo de configuración del teclado para ser usado por el Sistema X Window?

3. En una instalación típica de Linux un usuario puede cambiar a una terminal virtual presionando las teclas `Ctrl` + `Alt` + `F1`-`F6` en un teclado. Se le ha pedido que configure un sistema de kiosco con una interfaz gráfica y necesita que esta función esté desactivada para evitar la manipulación no autorizada del sistema. Usted decide crear un archivo de configuración `/etc/X11/xorg.conf.d/10-kiosk.conf`. Utilizando una sección `ServerFlags` (que se utiliza para establecer las opciones globales de Xorg en el servidor). ¿Qué opción habría que especificar? Revise la `página del manual de Xorg(1)` para buscar la opción.

Resumen

Esta lección cubrió el sistema X Window y su funcionamiento en Linux. El Sistema X Window se utiliza para dibujar imágenes y texto en las pantallas, tal y como se definen en varios archivos de configuración. El Sistema X Window se usa a menudo para configurar dispositivos de entrada como mouse y teclados. Esta lección trató los siguientes puntos:

- La arquitectura del Sistema X Window a un alto nivel.
- ¿Qué archivos de configuración se usan para configurar un Sistema X Window, y su ubicación en el sistema de archivos?
- ¿Cómo usar la variable de entorno `DISPLAY` en un sistema X?
- Una breve introducción al protocolo de visualización de Wayland.

Los comandos y archivos de configuración tratados fueron:

- Modificación de la disposición de un teclado dentro de una instalación Xorg con `setxkbmap` y `localectl`.
- El comando `Xorg` para crear un nuevo archivo de configuración `etc/X11/xorg.conf`.
- El contenido de los archivos de configuración de Xorg ubicados en: `/etc/X11/xorg.conf`, `/etc/X11/xorg.conf.d/` y `/usr/share/X11/xorg.conf.d/`.
- El comando `xdpyinfo` para mostrar información general sobre una sesión de servidor X en ejecución.

Respuesta a los ejercicios guiados

1. ¿Qué comando utilizaría para determinar qué extensiones Xorg están disponibles en un sistema?

```
$ xdpyinfo
```

2. Acaba de recibir un nuevo mouse de 10 botones para su computadora, sin embargo, requerirá una configuración extra para que todos los botones funcionen correctamente. Sin modificar el resto de la configuración del servidor X. ¿Qué directorio utilizaría para crear un nuevo archivo de configuración para este mouse y qué sección de configuración específica se utilizaría en este archivo?

   Las configuraciones definidas por el usuario deben estar ubicadas en `/etc/X11/xorg.conf.d/` y la sección específica necesaria para esta configuración del mouse sería `InputDevice`.

3. ¿Qué componente de una instalación de Linux es responsable de mantener un servidor X funcionando?

   El administrador de pantalla (Display manager).

4. ¿Qué opción en la línea de comandos se utiliza con el comando `X` para crear un nuevo archivo de configuración `xorg.conf`?

```
-configure
```

   Recuerde que el comando `X` es un enlace simbólico al comando `Xorg`.

Respuestas a los ejercicios de exploración

1. ¿Cuál sería el contenido de la variable de entorno `DISPLAY` en un sistema llamado `lab01` con una configuración de pantalla única? Supongamos que la variable de entorno `DISPLAY` se visualiza en un emulador de terminal en la tercera pantalla independiente.

```
$ echo $DISPLAY
lab01:0.2
```

2. ¿Qué comando se puede usar para crear un archivo de configuración del teclado para ser usado por el Sistema X Window?

```
$ localectl
```

3. En una instalación típica de Linux un usuario puede cambiar a una terminal virtual presionando las teclas `Ctrl` + `Alt` + `F1` - `F6` en un teclado. Se le ha pedido que configure un sistema de kiosco con una interfaz gráfica y necesita que esta función esté desactivada para evitar la manipulación no autorizada del sistema. Usted decide crear un archivo de configuración `/etc/X11/xorg.conf.d/10-kiosk.conf`. Utilizando una sección `ServerFlags` (que se utiliza para establecer las opciones globales de Xorg en el servidor). ¿Qué opción habría que especificar? Revise la `página del manual de Xorg(1)` para buscar la opción.

```
Section "ServerFlags"
    Option  "DontVTSwitch"  "True"
EndSection
```



106.2 Escritorios gráficos

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 106.2

Importancia

1

Áreas de conocimiento clave

- Conocer los principales entornos de escritorio.
- Conocer qué protocolos se utilizan para acceder a sesiones de escritorio remoto.

Lista parcial de archivos, términos y utilidades

- KDE
- Gnome
- Xfce
- X11
- XDMCP
- VNC
- Spice
- RDP



106.2 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	106 Interfaces de usuario y escritorios
Objetivo:	106.2 Escritorios gráficos
Lección:	1 de 1

Introducción

Los sistemas operativos basados en Linux son conocidos por su avanzada interfaz de línea de comandos, pero puede ser intimidante para los usuarios no técnicos. Con la intención de hacer la computadora más intuitiva, la combinación de pantallas de alta resolución, dio nacimiento a interfaces de usuario impulsadas por imágenes. Mientras que la interfaz de línea de comandos requiere un conocimiento previo de los nombres de los programas y sus opciones de configuración, con una *interfaz gráfica de usuario "Graphical User Interface"* (GUI) la funcionalidad del programa puede activarse apuntando a elementos visuales, haciendo que la curva de aprendizaje sea más plana. Además, la interfaz gráfica es más adecuada para actividades multimedia y otras actividades de orientación visual.

De hecho, la interfaz gráfica de usuario es casi un sinónimo de interfaz de escritorio y la mayoría de las distribuciones de Linux vienen con una instalada por defecto. Sin embargo, no hay un solo programa monolítico responsable de los escritorios gráficos completos disponibles en los sistemas Linux. En su lugar, cada escritorio gráfico es de hecho una gran colección de programas y sus dependencias, que varían desde las elecciones de la distribución hasta el gusto personal de cada usuario.

Sistema X Window

En Linux y otros sistemas operativos similares a Unix en los que se emplea, el *sistema X Window* (conocido como *X11* o simplemente *X*) proporciona los recursos de bajo nivel relacionados con la representación de la interfaz gráfica y la interacción del usuario con ella. Por ejemplo:

- El manejo de los eventos de entrada, como los movimientos del mouse o las pulsaciones de teclas.
- La capacidad de cortar, copiar y pegar el contenido del texto entre aplicaciones separadas.
- La interfaz de programación que otros programas utilizan para dibujar los elementos gráficos.

Aunque el sistema X Windows se encarga de controlar la pantalla gráfica (el controlador de vídeo en sí es parte de X), no pretende dibujar elementos visuales complejos. Las formas, los colores, los matices y cualquier otro efecto visual son generados por la aplicación que se ejecuta sobre X. Este enfoque da a las aplicaciones mucho espacio para crear interfaces personalizadas, pero también puede dar lugar a sobreusos de desarrollo que están fuera del alcance de la aplicación y a incoherencias en el aspecto y comportamiento cuando se comparan con las interfaces de otros programas.

Desde el punto de vista del desarrollador, la introducción del *ambiente de escritorio* facilita la programación de la interfaz gráfica de usuario vinculada al desarrollo de la aplicación subyacente, mientras que desde el punto de vista del usuario proporciona una experiencia coherente entre las distintas aplicaciones. Los entornos de escritorio reunen interfaces de programación, bibliotecas y programas de apoyo que cooperan para ofrecer conceptos de diseño tradicionales pero aún en evolución.

Ambientes de escritorio

La tradicional interfaz gráfica de la computadora de escritorio consiste en varias ventanas — el término *ventana* se usa aquí para referirse a cualquier área de la pantalla autónoma — asociada con procesos en ejecución. Como el sistema X Window por sí solo ofrece sólo características interactivas básicas, la experiencia completa del usuario depende de los componentes proporcionados por el entorno de escritorio. Probablemente el componente más importante de un entorno de escritorio, es el *administrador de ventanas (windows manager)* que controla la colocación y decoración de las ventanas. Es el gestor de ventanas que añade la barra de título a la ventana, los botones de control — generalmente asociados con las acciones de minimizar, maximizar y cerrar — y gestiona el cambio entre las ventanas abiertas.

> **NOTE**
>
> Los conceptos básicos que se encuentran en las interfaces gráficas de las computadoras de escritorio provienen de ideas tomadas de los espacios de trabajo reales de oficinas. Hablando metafóricamente, la pantalla de la computadora es el escritorio, donde se colocan objetos como documentos y carpetas. Una ventana de aplicación con el contenido de un documento imita los actos físicos como rellenar un formulario o pintar un cuadro. Como verdaderos escritorios, los escritorios de las computadoras también tienen accesorios de software como bloc de notas, relojes, calendarios, etc., la mayoría de ellos basados en sus contrapartes "reales".

Todos los entornos de escritorio proporcionan un gestor de ventanas que se ajusta al aspecto y a la sensación de su *kit de herramientas de widgets*. Los widgets son elementos visuales informativos o interactivos, como botones o campos de entrada de texto, distribuidos dentro de la ventana de la aplicación. Los componentes estándares del escritorio, — como el lanzador de aplicaciones, la barra de tareas, etc. — y el propio gestor de ventanas dependen de tales kits de herramientas de widgets para ensamblar sus interfaces.

Las bibliotecas de software, como *GTK+* y *Qt*, proporcionan widgets que los programadores pueden usar para construir elaboradas interfaces gráficas para sus aplicaciones. Históricamente, las aplicaciones desarrolladas con GTK+ no se parecían a las aplicaciones hechas con Qt y viceversa, pero el mejor soporte de temas de los entornos de escritorio de hoy en día hacen que la distinción sea menos obvia.

En general, GTK+ y Qt ofrecen las mismas características en cuanto a los widgets. Los elementos interactivos simples pueden ser indistinguibles, mientras que los widgets compuestos, como la ventana de diálogo que utilizan las aplicaciones para abrir o guardar archivos, sin embargo, pueden tener un aspecto bastante diferente. No obstante, las aplicaciones construidas con conjuntos de herramientas distintos pueden ejecutarse simultáneamente, independientemente del conjunto de herramientas de widgets utilizado por los demás componentes del escritorio.

Además de los componentes básicos del escritorio, que podrían considerarse programas individuales por sí mismos, los entornos de escritorio persiguen la metáfora del escritorio proporcionando un conjunto mínimo de accesorios desarrollados bajo las mismas pautas de diseño. Las variaciones de las siguientes aplicaciones son comúnmente proporcionadas por todos los principales entornos de escritorio:

Aplicaciones relacionadas con el sistema

Emulador de terminales, gestor de archivos, gestor de instalación de paquetes, herramientas de configuración del sistema.

Comunicación e Internet
Administrador de contactos, cliente de correo electrónico, navegador web.

Aplicaciones de oficina
Calendario, calculadora, editor de texto.

Los entornos de escritorio pueden incluir muchos otros servicios y aplicaciones: el saludo de la pantalla de inicio de sesión, el gestor de sesiones, la comunicación entre procesos, el agente de credenciales, etc. También incorporan características proporcionadas por servicios de sistemas de terceros, como *PulseAudio* para el sonido y *CUPS* para la impresión. Estas características no necesitan el entorno gráfico para funcionar, pero el entorno de escritorio proporciona gráficos frontend para facilitar la configuración y el funcionamiento de esos recursos.

Entornos de escritorio populares

Muchos sistemas operativos patentados sólo admiten un único entorno oficial de escritorio que está vinculado a su lanzamiento particular y que se supone que no debe ser modificado. A diferencia de ellos, los sistemas operativos basados en Linux soportan diferentes opciones de entornos de escritorio que pueden utilizarse en conjunto con X. Cada entorno de escritorio tiene sus propias características, pero normalmente comparten algunos conceptos de diseño comunes:

- Un lanzador (launcher) de aplicaciones que lista las aplicaciones integradas y de terceros disponibles en el sistema.
- Reglas que definen las aplicaciones predeterminadas asociadas a los tipos de archivos y protocolos.
- Herramientas de configuración para personalizar la apariencia y el comportamiento del entorno de escritorio.

Gnome es uno de los entornos de escritorio más populares, siendo la primera opción en distribuciones como Fedora, Debian, Ubuntu, SUSE Linux Enterprise, Red Hat Enterprise Linux, CentOS, etc. En su versión 3, Gnome trajo grandes cambios en su aspecto y estructura, alejándose de la metáfora del escritorio e introduciendo el *Gnome Shell* como su nueva interfaz. .Gnome Shell Activities image::./images/gnome-debian-10.1.0.png[pdfwidth="75%"]

El lanzador (launcher) de pantalla completa *Gnome Shell Activities* reemplazó al tradicional lanzador de aplicaciones y a la barra de tareas. Sin embargo, todavía es posible usar Gnome 3 con el aspecto antiguo eligiendo la opción *Gnome Classic* en la pantalla de inicio de sesión.

KDE es un gran ecosistema de aplicaciones y plataforma de desarrollo. Su última versión de entorno de escritorio, *KDE Plasma*, se utiliza por defecto en openSUSE, Mageia, Kubuntu, etc. El empleo de la

biblioteca Qt es la característica más destacada de KDE, que le da su aspecto inconfundible y una plétora de aplicaciones originales. KDE incluso proporciona una herramienta de configuración para asegurar la cohesión visual con las aplicaciones GTK+.

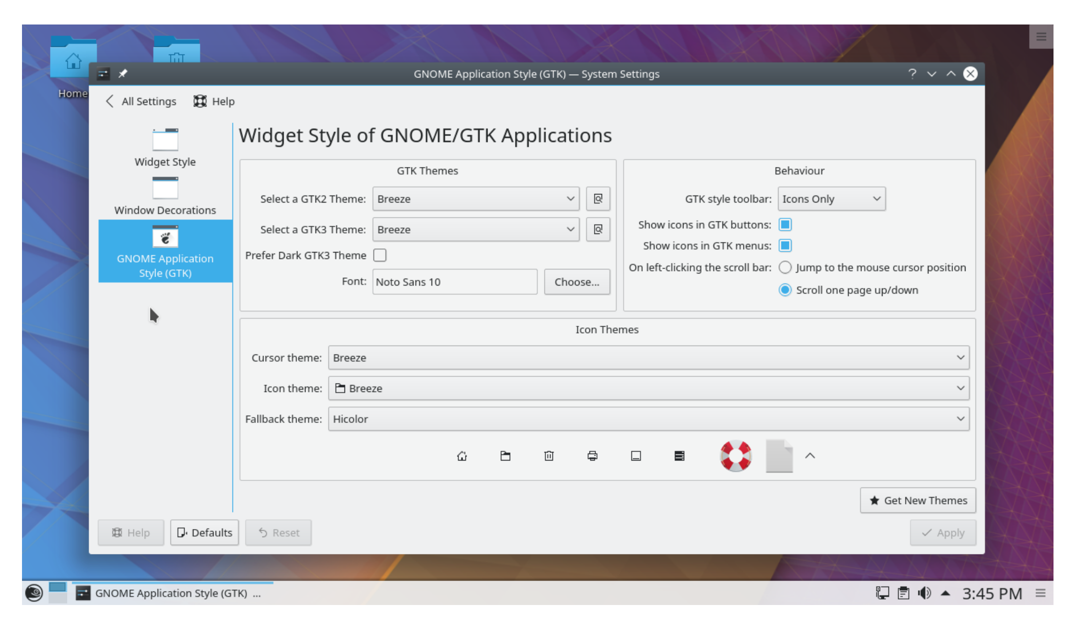


Figure 1. Configuraciones KDE GTK

Xfce es un entorno de escritorio que pretende ser estéticamente agradable sin consumir muchos recursos de la máquina. Su estructura está altamente modularizada, permitiendo al usuario activar y desactivar componentes según sus necesidades y preferencias.

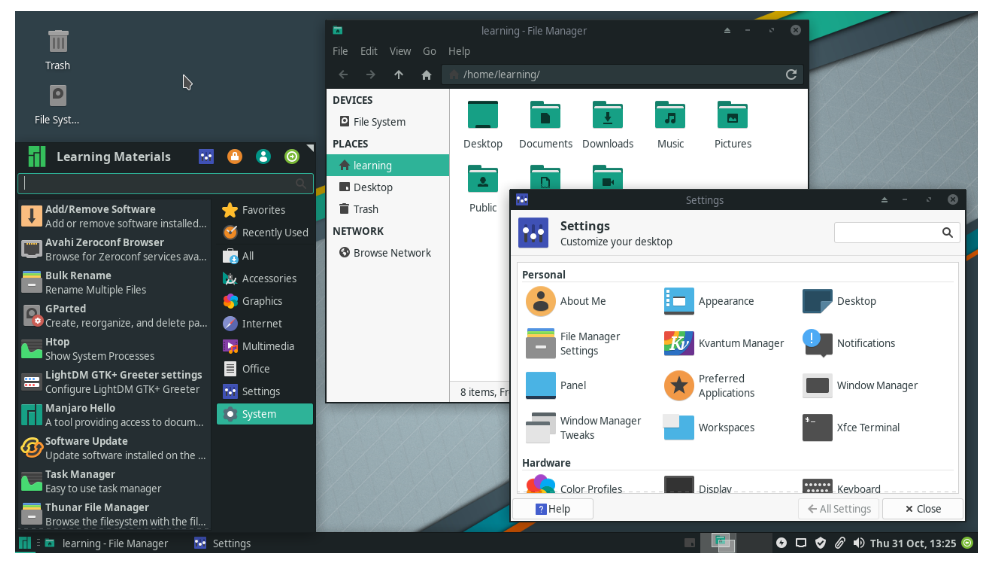


Figure 2. Escritorio de Xfce

Hay muchos otros entornos de escritorio para Linux, normalmente proporcionados por bifurcaciones de distribuciones alternativas. La distribución Linux Mint, por ejemplo, proporciona dos entornos de

escritorio originales: *Cinnamon* (un bifurcación de Gnome 3) y *MATE* (un bifurcación de Gnome 2). *LXDE* es un entorno de escritorio adaptado al bajo consumo de recursos, lo que lo convierte en una buena opción para su instalación en equipos antiguos o en ordenadores monoplaca. Aunque no ofrece todas las características de los entornos de escritorio más pesados, LXDE ofrece todas las características básicas que se esperan de una moderna interfaz gráfica de usuario.

TIP Los atajos o shortcuts de teclado juegan un papel importante en la interacción con el entorno del escritorio. Algunos atajos de teclado, como `Alt`+`Tab` para cambiar entre ventanas o `Ctrl`+`c` para copiar texto, son universales en todos los entornos de escritorio, pero cada entorno de escritorio también tiene sus propios shortcuts. Los atajos de teclado se encuentran en la herramienta de configuración de teclado que proporciona el entorno de escritorio, donde se pueden añadir o modificar los shortcuts.

Interoperabilidad de escritorio

La diversidad de entornos de escritorio en los sistemas operativos basados en Linux impone un reto: ¿cómo hacer que funcionen correctamente con aplicaciones gráficas o servicios de sistema de terceros sin tener que implementar un soporte específico para cada uno de ellos? Los métodos y especificaciones compartidos entre entornos de escritorio mejoran en gran medida la experiencia del usuario y resuelven muchos problemas de desarrollo, ya que las aplicaciones gráficas deben interactuar con el entorno de escritorio actual, independientemente del entorno de escritorio para el que fueron diseñadas originalmente. Además, es importante mantener la configuración general del escritorio si el usuario acaba cambiando su elección de entorno de escritorio.

La organización *freedesktop.org* mantiene un gran cuerpo de especificaciones para la interoperabilidad de los ordenadores de sobremesa. La adopción de la especificación completa no es obligatoria, pero muchas de ellas se utilizan ampliamente:

Ubicaciones de directorios
: Donde se encuentran los ajustes personales y otros archivos específicos del usuario.

Entradas en el escritorio
: Las aplicaciones de línea de comandos pueden ejecutarse en el entorno de escritorio a través de cualquier emulador de terminal, pero sería demasiado confuso hacer que todas ellas estuvieran disponibles en el lanzador de aplicaciones. Las entradas de escritorio son archivos de texto que terminan en `.desktop` que son utilizados por el entorno de escritorio para recopilar información sobre las aplicaciones de escritorio disponibles y cómo utilizarlas.

Aplicación de arranque automático

Entradas del escritorio que indican la aplicación que debe iniciarse automáticamente después de que el usuario haya iniciado la sesión.

Arrastrar y soltar

Cómo las aplicaciones deben manejar los eventos de arrastrar y soltar.

Papelera (trash can)

La ubicación común de los archivos eliminados por el administrador de archivos, así como los métodos para almacenar y eliminar archivos de allí.

Temas de iconos

El formato común de las bibliotecas de iconos intercambiables.

La facilidad de uso que ofrecen los entornos de escritorio tiene un inconveniente en comparación con las interfaces de texto como shell: la capacidad de proporcionar acceso remoto. Mientras que un entorno de shell de línea de comandos de un equipo remoto puede ser fácilmente accesible con herramientas como `ssh`, el acceso remoto a entornos gráficos requiere métodos diferentes y puede no lograr un rendimiento satisfactorio en conexiones más lentas.

Acceso no local

El Sistema X Window adopta un diseño basado en *pantallas* autónomas, donde el mismo *administrador de pantalla* puede controlar más de una sesión de escritorio gráfico al mismo tiempo. En esencia, una pantalla es análoga para un terminal de texto: ambos se refieren a una máquina o aplicación de software utilizada como punto de entrada para establecer una sesión de sistema operativo independiente. Aunque la configuración más común implica una sesión gráfica singular que se ejecuta en la máquina local, también son posibles otras configuraciones menos convencionales:

- Cambiar entre sesiones de escritorio gráfico activas en la misma máquina.
- Más de un conjunto de dispositivos de visualización (por ejemplo, pantalla, teclado, mouse) conectados a la misma máquina, cada uno controlando su propia sesión de escritorio gráfico.
- Sesiones remotas de escritorio gráfico, donde la interfaz gráfica se envía a través de la red a una pantalla remota.

Las sesiones de escritorio remoto son soportadas nativamente por X, que emplea el *X Display Manager Control Protocol* (XDMCP) para comunicarse con las pantallas remotas. Debido a su alto uso de ancho de banda, el XDMCP se utiliza raramente a través de Internet o en redes LAN de baja velocidad. Los problemas de seguridad también son una preocupación con XDMCP: la pantalla local

se comunica con un administrador de pantalla X remoto con privilegios para ejecutar procedimientos remotos, por lo que una eventual vulnerabilidad podría favorecer la ejecución de comandos arbitrarios con privilegios en el equipo remoto.

Además, el XDMCP requiere que se ejecuten X instancias en ambos extremos de la conexión, lo que puede hacerla inviable si el Sistema X Window no está disponible para todas las máquinas involucradas. En la práctica, se utilizan otros métodos más eficientes y menos invasivos para establecer sesiones de escritorio gráfico a distancia.

Virtual Network Computing (VNC) es una herramienta de plataforma independiente para ver y controlar entornos de escritorio remotos usando el protocolo *Remote Frame Buffer* (RFB). A través de este, los eventos producidos por el teclado y el mouse, se transmiten al escritorio remoto, que a su vez devuelven cualquier actualización de la pantalla para ser mostrada localmente. Es posible ejecutar muchos servidores VNC en la misma máquina, pero cada servidor VNC necesita un puerto TCP exclusivo en la interfaz de red que acepte las solicitudes de sesión entrantes. Por convención, el primer servidor de VNC debe usar el puerto TCP 5900, el segundo debe usar el 5901, y así sucesivamente.

El servidor VNC no necesita privilegios especiales para funcionar. Por ejemplo, un usuario ordinario puede iniciar sesión en su cuenta remota e iniciar su propio servidor VNC desde allí. Luego, en la máquina local puede utilizar cualquier aplicación cliente de VNC para acceder al escritorio remoto (suponiendo que se pueda acceder a los puertos de red correspondientes). El archivo `~/.vnc/xstartup` es un script de shell ejecutado por el servidor VNC cuando se inicia y puede ser utilizado para definir qué entorno de escritorio, el servidor VNC pondrá a disposición del cliente VNC. Es importante señalar que VNC no proporciona métodos modernos de cifrado y autenticación de forma nativa, por lo que debe utilizarse junto con una aplicación de terceros que proporcione esas características. Los métodos que implican túneles de VPN y SSH se utilizan a menudo para asegurar las conexiones de VNC.

Remote Desktop Protocol (RDP) se utiliza principalmente para acceder de forma remota al escritorio de un sistema operativo *Microsoft Windows* a través del puerto de red TCP 3389. Aunque utiliza el protocolo RDP de Microsoft, la implementación cliente utilizada en los sistemas Linux son programas de código abierto licenciados bajo *GNU General Public License* (GPL) y no tienen restricciones legales de uso.

Simple Protocol for Independent Computing Environments (Spice) comprende un conjunto de herramientas destinadas para acceder al entorno de escritorio de los sistemas *virtualizados*, ya sea en la máquina local o en una ubicación remota. Además, el protocolo Spice ofrece características nativas para integrar los sistemas locales y remotos como la posibilidad de acceder a los dispositivos locales (por ejemplo, los altavoces de sonido y los dispositivos USB conectados) desde la máquina remota y el intercambio de archivos entre los dos sistemas.

Hay comandos específicos del cliente para conectarse a cada uno de estos protocolos de escritorio remoto, pero el cliente de escritorio remoto *Remmina* proporciona una interfaz gráfica integrada que facilita el proceso de conexión, almacenando opcionalmente la configuración de la conexión para su uso posterior. Remmina tiene plugins para cada protocolo individual y hay plugins para XDMCP, VNC, RDP y Spice. La elección de la herramienta adecuada depende de los sistemas operativos implicados, la calidad de la conexión de red y las características del entorno de escritorio remoto que deben estar disponibles.

Guided Exercises

1. ¿Qué tipo de aplicación proporciona sesiones de shell con ventanas en el entorno de escritorio?

2. Debido a la variedad de entornos de escritorio de Linux, la misma aplicación puede tener más de una versión, cada una de ellas más adecuada para un conjunto de herramientas de widgets en particular. Por ejemplo, el cliente bittorrent *Transmission* tiene dos versiones: *transmission-gtk* y *transmission-qt*. ¿Cuál de las dos debe ser instalada para asegurar la máxima integración con KDE?

3. ¿Qué entornos de escritorio Linux se recomiendan para equipos monoplaca de bajo costo con poca potencia de procesamiento?

Explorational Exercises

1. Hay dos maneras de copiar y pegar texto en el sistema X WIndow: usando las tradicionales teclas `Ctrl` + `c` y `Ctrl` + `v` (también disponibles en el menú de la ventana) o usar el clic del botón central del ratón para pegar el texto actualmente seleccionado. ¿Cuál es la manera apropiada de copiar y pegar texto de un emulador de terminal?

2. La mayoría de los entornos de escritorio asignan el "atajo" en la combinación `Alt` + `F2` a la ventana *Ejecutar programa*, donde los programas pueden ser ejecutados en forma de línea de comandos. En KDE, ¿Qué comando ejecutaría el emulador de terminal por defecto?

3. ¿Qué protocolo es el más adecuado para acceder a un escritorio remoto de Windows desde un entorno de escritorio de Linux?

Resumen

Esta lección es una visión general de los escritorios gráficos disponibles para los sistemas Linux. El sistema X Window por sí solo proporciona sólo características de interfaz simples, por lo que los entornos de escritorio extienden la experiencia del usuario en la interfaz gráfica con ventanas. La lección abarca los siguientes temas:

- Interfaz gráfica y conceptos del Sistema X Window.
- Entornos de escritorio disponibles para Linux.
- Similitudes y diferencias entre los entornos de escritorio.
- ¿Cómo acceder a un entorno de escritorio remoto?

Los conceptos y programas abordados fueron:

- SIstema X Window
- Entornos de escritorio populares: KDE, Gnome, Xfce.
- Protocolos de acceso remoto: XDMCP, VNC, RDP, Spice.

Respuesta a los ejercicios guiados

1. ¿Qué tipo de aplicación proporciona sesiones de shell con ventanas en el entorno de escritorio?

   Cualquier emulador de terminal como Konsole, terminal Gnome, xterm, etc., dará acceso a una sesión de shell interactiva local.

2. Debido a la variedad de entornos de escritorio de Linux, la misma aplicación puede tener más de una versión, cada una de ellas más adecuada para un conjunto de herramientas de widgets en particular. Por ejemplo, el cliente bittorrent *Transmission* tiene dos versiones: *transmission-gtk* y *transmission-qt*. ¿Cuál de las dos debe ser instalada para asegurar la máxima integración con KDE?

   KDE está construido sobre la biblioteca Qt, así que la versión Qt — *transmission-qt* — debe ser instalada.

3. ¿Qué entornos de escritorio Linux se recomiendan para equipos monoplaca de bajo costo con poca potencia de procesamiento?

   Entornos de escritorio básicos que no usan demasiados efectos visuales, como Xfce y LXDE.

Respuestas a los ejercicios de exploración

1. Hay dos maneras de copiar y pegar texto en el sistema X WIndow: usando las tradicionales teclas `Ctrl` + `c` y `Ctrl` + `v` (también disponibles en el menú de la ventana) o usar el clic del botón central del ratón para pegar el texto actualmente seleccionado. ¿Cuál es la manera apropiada de copiar y pegar texto de un emulador de terminal?

   Las sesiones de shell interactivas asignan la tecla `Ctrl` + `c` para detener la ejecución del programa, por lo que se recomienda el método del botón medio.

2. La mayoría de los entornos de escritorio asignan el "atajo" en la combinación `Alt` + `F2` a la ventana *Ejecutar programa*, donde los programas pueden ser ejecutados en forma de línea de comandos. En KDE, ¿Qué comando ejecutaría el emulador de terminal por defecto?

   El comando `konsole` ejecuta el emulador de terminales de KDE, pero los términos genéricos como *terminal* también funcionan.

3. ¿Qué protocolo es el más adecuado para acceder a un escritorio remoto de Windows desde un entorno de escritorio de Linux?

   El Protocolo de Escritorio Remoto (RDP), tal y como lo soportan nativamente tanto Windows como Linux.



106.3 Accesibilidad

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 106.3

Importancia

1

Áreas de conocimiento clave

- Conocimientos básicos de temas y configuraciones visuales.
- Conocimientos básicos de tecnología asistida.

Lista parcial de archivos, términos y utilidades

- Temas de escritorio de alto contraste/impresión grande.
- Lector de pantalla.
- Pantalla braille.
- Lupa de pantalla.
- Teclado en pantalla.
- Teclas pegajosas/de repetición.
- Teclas lentas/de rebote/de conmutación.
- Teclas de ratón.
- Gestos.
- Reconocimiento de voz.



106.3 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	106 Interfaces de usuario y escritorios
Objetivo:	106.3 Accesibilidad
Lección:	1 de 1

Introducción

El entorno de escritorio de una distribución Linux tiene muchas configuraciones y herramientas para adaptar la interfaz de usuario a personas con discapacidades. Los dispositivos ordinarios con interfaz como pantalla, teclado y mouse/touchpad pueden ser reconfigurados individualmente para mejorar las deficiencias visuales o la movilidad reducida.

Por ejemplo, es posible ajustar la combinación de colores del escritorio para que las personas daltónicas mantengan una mejor visualización. Además, las personas que sufren lesiones por esfuerzo repetitivo pueden aprovechar los métodos alternativos de mecanografía y apunte.

Algunas de estas características de accesibilidad son proporcionadas por el propio entorno de escritorio, como Gnome o KDE, y otras son proporcionadas por programas adicionales. En este último caso, es importante elegir la herramienta que mejor integra el entorno de escritorio, para que la ayuda sea de mejor calidad.

Configuración de accesibilidad

Todas las distribuciones principales de Linux proporcionan características similares de accesibilidad y estas pueden ser personalizadas con un módulo de configuración que se encuentra en el

administrador de configuración proporcionado por el entorno de escritorio. En el escritorio de Gnome, el módulo de configuración de accesibilidad se llama *Acceso universal*, mientras que en KDE se encuentra en *Configuración del sistema*, *Personalización*, *Accesibilidad*. Otros entornos de escritorio, como *Xfce*, también lo llaman *Accesibilidad* y se encuentra en su administrador de ajustes gráficos. Sin embargo, ofrecen un conjunto reducido de funcionalidades en comparación con Gnome y KDE.

Gnome puede ser configurado para mostrar permanentemente el menú de Acceso Universal en la esquina superior derecha de la pantalla, donde las opciones de accesibilidad pueden ser rápidamente activadas. Por ejemplo, puede utilizarse para sustituir la alerta sonora por una visual, de modo que los usuarios con problemas de audición puedan percibir las alertas del sistema con mayor facilidad. Aunque KDE no tiene un menú de acceso rápido similar, la función de alerta visual también está disponible, pero en su lugar se llama *campana visual*.

Asistencia de teclado y mouse

El comportamiento del teclado y el mouse puede ser modificado para evitar dificultades específicas de movilidad. Las combinaciones de teclas, la tasa de repetición automática y las pulsaciones involuntarias de teclas pueden ser obstáculos importantes para los usuarios con movilidad reducida en sus manos. Estas deficiencias de mecanografía se abordan mediante tres características de accesibilidad relacionadas con el teclado: *Teclas adhesivas (sticky keys)*, *Teclas de rebote (Bounce keys)* y *Teclas lentas (slow keys)*.

La función *Teclas adhesivas*, que se encuentra en la sección *Asistente de digitación* de la configuración de acceso universal de Gnome, permite al usuario teclear atajos con una tecla a la vez. Cuando esta función se activa, las combinaciones de teclas como `Ctrl`+`C` no necesitan ser presionadas al mismo tiempo. El usuario puede pulsar primero la tecla `Ctrl`, soltarla y luego pulsar la tecla `C`. En KDE, esta opción se encuentra en la pestaña *Teclas de modificación* de la ventana de configuración de accesibilidad. KDE también ofrece la opción *Teclas de bloqueo*: si se encuentra activada, las teclas `Alt`, `Ctrl` y Shift permanecerán “abajo”, si el usuario las presiona dos veces, el comportamiento será similar a la tecla de bloqueo de mayúsculas. Al igual que esta función, el usuario deberá pulsar nuevamente la tecla correspondiente y soltarla.

La función *Teclas de rebote* trata de inhibir las pulsaciones de las teclas no intencionadas agregando un retardo entre ellas, es decir, una nueva pulsación de tecla será aceptada sólo después de que haya transcurrido un tiempo determinado desde la última pulsación de la misma. Los usuarios con temblores en las manos pueden encontrar útil la función *Teclas de rebote* para evitar pulsar una tecla varias veces cuando sólo tienen intención de pulsarla una vez. En Gnome, esta característica sólo afecta a las mismas repeticiones de teclas, mientras que en KDE afecta a cualquier otra pulsación de tecla. Además esta función se encuentra en la pestaña *Filtros de teclado*.

La característica de las *teclas lentas* también ayuda a evitar las pulsaciones accidentales. Cuando se encuentra activada, las teclas lentas requerirán que el usuario mantenga pulsada la tecla durante un tiempo determinado antes de ser aceptada. Según las necesidades del usuario, también puede ser útil ajustar la repetición automática mientras se mantiene pulsada una tecla (disponible en la configuración del teclado).

Las funciones de accesibilidad de las teclas adhesivas y lentas se pueden activar y desactivar con *Gestos de activación* realizados en el teclado. En KDE, la opción *Utilizar gestos para activar las teclas adhesivas y las teclas lentas* debe estar marcada para activar dichos gestos, mientras que en Gnome esta característica se llama *Habilitar por teclado* en la ventana de configuración *Asistente de digitación*. Una vez que los Gestos de Activación están habilitados, las características de las teclas adhesivas se activarán después de presionar la tecla Shift cinco veces consecutivas. Para activar la función de las teclas de desaceleración, la tecla Shift debe mantenerse pulsada durante ocho segundos.

Aquellos usuarios que encuentran más cómodo el uso del teclado sobre el mouse o el touchpad pueden recurrir a los atajos de teclado para movilizarse en el entorno del escritorio. Además, una función llamada *Teclas del mouse* permite al usuario controlar el puntero del mouse con el teclado numérico. Esta opción está presente en los teclados de escritorio de tamaño completo y en las computadoras portátiles de mayor tamaño.

El teclado numérico está dispuesto en una cuadrícula, de modo que cada número corresponde a una dirección: *2* mueve el cursor hacia abajo, *4* mueve el cursor hacia la izquierda, *7* mueve el cursor hacia el noroeste, etc. Por defecto, el número *5* corresponde al clic izquierdo del ratón.

Mientras que en Gnome sólo hay una opción para habilitar la función de Teclas del mouse en la ventana de ajustes de Acceso Universal, en KDE los ajustes de las Teclas del mouse se encuentran en *Configuración del sistema*, *Mouse* y *Navegación por teclado*, también las opciones como la velocidad y la aceleración pueden ser personalizadas.

TIP

Las teclas lentas, las adhesivas, las de rebote y las teclas de mouse son características de accesibilidad proporcionadas por AccessX, un recurso dentro de la extensión del teclado X del Sistema X Window. Los ajustes de AccessX también pueden ser modificados desde la línea de comandos, con el comando "xkbset".

El mouse o el touchpad pueden utilizarse para generar la entrada del teclado cuando el uso del mismo es demasiado molesto o no es posible. Si se activa la opción *Teclado de pantalla* en la configuración de acceso universal de Gnome, aparecerá un teclado en pantalla cada vez que el cursor esté en un campo de texto y se introduzca un nuevo texto haciendo clic en las teclas con el mouse o en la pantalla táctil, de forma muy parecida al teclado virtual de los smartphones.

Es posible, que por defecto, KDE y otros entornos de escritorio no proporcionen el teclado de pantalla, pero el paquete *onboard* puede ser instalado manualmente para proporcionar un simple teclado en pantalla que puede ser utilizado en cualquier entorno de escritorio. Después de la instalación, estará disponible como una aplicación regular en el lanzador de aplicaciones.

El comportamiento del puntero también puede modificarse si al hacer clic y mover el mouse se produce dolor o no es práctico por cualquier otra razón. Si por ejemplo, el usuario no es capaz de hacer clic en el botón del mouse con la rapidez suficiente para desencadenar un evento de doble clic, el intervalo de tiempo para pulsar el botón del mouse por segunda vez para hacer doble clic puede aumentarse en las *Preferencias del mouse* en la ventana de configuración del sistema.

Si el usuario no es capaz de pulsar uno de los botones del mouse o ninguno de ellos, entonces los clics del mouse pueden ser simulados mediante diferentes técnicas. En la sección *Asistente de click* del *Acceso universal de Gnome*, la opción *Simular un clic con el botón derecho del ratón* generará un clic derecho si el usuario presiona y mantiene el botón izquierdo del ratón. Con la opción *Simular un clic flotante* activada, se generará un evento de clic cuando el usuario mantenga el mouse sin movimiento. En KDE, la aplicación *KMouseTool* proporcionará estas mismas características para ayudar a las acciones del mouse.

Deficiencias visuales

Los usuarios con visión reducida pueden seguir utilizando la pantalla del monitor para interactuar con la computadora. Dependiendo de las necesidades del usuario, se pueden hacer muchos ajustes visuales para mejorar los detalles del escritorio gráfico, que de otra manera serían difíciles de observar.

La sección *Ver* de los ajustes de *Acceso Universal de Gnome* proporciona opciones que pueden ayudar a las personas con visión reducida:

Alto contraste

Hará que las ventanas y los botones sean más fáciles de ver dibujándolos en colores más nítidos.

Texto grande

Ampliará el tamaño de la fuente en la pantalla.

Tamaño del cursor

Permite elegir un cursor del mouse más grande, facilitando su localización en la pantalla.

Algunos de estos ajustes no están estrictamente relacionados con las características de accesibilidad, por lo que pueden encontrarse en la sección de apariencia de la utilidad de configuración proporcionada por otros entornos de escritorio. Un usuario que posea dificultades para discernir

entre los elementos visuales puede elegir un tema de alto contraste para facilitar la identificación de los botones, las ventanas, etc.

Si los ajustes de apariencia por sí solos no son suficientes para mejorar la legibilidad de la pantalla, entonces se puede utilizar un programa de ampliación de pantalla para acercar partes de la misma. Esta característica se llama *Zoom* y se encuentra en los ajustes de *Acceso Universal de Gnome*, donde se pueden personalizar opciones como la relación de aumento, la posición de la lupa y los ajustes de color.

En KDE, el programa *KMagnifier* proporciona las mismas características, pero está disponible como una aplicación ordinaria a través del lanzador de aplicaciones. Otros entornos de escritorio pueden proporcionar sus propios magnificadores de pantalla. Por ejemplo, Xfce acercará y alejará la pantalla girando la rueda de desplazamiento del ratón mientras la tecla `Alt` está presionada.

Por último, los usuarios para los que la interfaz gráfica no es una opción, pueden utilizar un *lector de pantalla* para interactuar con la computadora. Independientemente del entorno de escritorio elegido, el lector de pantalla más popular para los sistemas Linux es *Orca*, que suele instalarse por defecto en la mayoría de las distribuciones. Orca genera una voz sintetizada para reportar eventos de pantalla y para leer el texto bajo el cursor del mouse. Orca también trabaja con *refreshable braille displays*, dispositivos especiales que muestran los caracteres braille levantando pequeños alfileres que se pueden sentir con la punta de los dedos. No todas las aplicaciones de escritorio están totalmente adaptadas a los lectores de pantalla y no todos los usuarios los encontrarán fáciles de usar, pero lo importante es proporcionar el mayor número posible de estrategias de lectura de pantalla a elegir.

Guided Exercises

1. ¿Qué característica de accesibilidad podría ayudar a un usuario a alternar entre las ventanas abiertas utilizando el teclado, teniendo en cuenta que el usuario no puede pulsar las teclas `Alt` y `Tab` al mismo tiempo?

2. ¿Cómo podría la función de accesibilidad de *teclas de rebote* ayudar a los usuarios cuyos temblores (involuntarios) en las manos dificultan su escritura?

3. ¿Cuáles son los gestos de activación más comunes para la función de accesibilidad de las *teclas adhesivas*?

Explorational Exercises

1. Las características de accesibilidad pueden no ser proporcionadas por una sola aplicación y pueden variar de un entorno de escritorio a otro. En KDE, ¿Qué aplicación ayuda a los que tienen lesiones por esfuerzo repetitivo haciendo clic en el mouse cuando el cursor del mismo se detiene brevemente?

2. ¿Qué aspectos de la apariencia del entorno gráfico pueden modificarse para facilitar la lectura del texto en la pantalla?

3. ¿De qué manera puede la aplicación *Orca* ayudar a los usuarios con problemas de visión a interactuar con el entorno de escritorio?

Resumen

Esta lección cubre las características de accesibilidad general disponibles en los sistemas Linux. Todos los principales entornos de escritorio, especialmente Gnome y KDE, proporcionan muchas aplicaciones incorporadas y de terceros para ayudar a las personas con discapacidades visuales o movilidad reducida. La lección abarca los siguientes temas:

- Cómo cambiar la configuración de accesibilidad.
- Formas alternativas de usar el teclado y el ratón.
- Mejoras en el escritorio para los discapacitados visuales.

Los comandos y procedimientos abordados fueron:

- Configuración de accesibilidad del teclado: Teclas adhesivas, lentas y de rebote.
- Generar artificialmente eventos de mouse.
- Teclado en pantalla.
- Ajustes visuales para mejorar la legibilidad.
- Temas de escritorio de alto contraste y de tamaño de la fuente.
- Lupas de pantalla.
- El lector de pantalla Orca.

Respuesta a los ejercicios guiados

1. ¿Qué característica de accesibilidad podría ayudar a un usuario a alternar entre las ventanas abiertas utilizando el teclado, teniendo en cuenta que el usuario no puede pulsar las teclas `Alt` y `Tab` al mismo tiempo?

   La característica de las teclas adhesivas permite al usuario escribir atajos de una tecla a la vez.

2. ¿Cómo podría la función de accesibilidad de *teclas de rebote* ayudar a los usuarios cuyos temblores (involuntarios) en las manos dificultan su escritura?

   Con las teclas de rebote habilitadas, una nueva pulsación de tecla se aceptará sólo después de que haya pasado un tiempo determinado desde la última pulsación de la misma.

3. ¿Cuáles son los gestos de activación más comunes para la función de accesibilidad de las *teclas adhesivas*?

   Si se activan los gestos de activación, la función de las teclas adhesivas se activará después de pulsar la tecla `Shift` cinco veces consecutivas.

Respuestas a los ejercicios de exploración

1. Las características de accesibilidad pueden no ser proporcionadas por una sola aplicación y pueden variar de un entorno de escritorio a otro. En KDE, ¿Qué aplicación ayuda a los que tienen lesiones por esfuerzo repetitivo haciendo clic en el mouse cuando el cursor del mismo se detiene brevemente?

   La aplicación *KMouseTool*.

2. ¿Qué aspectos de la apariencia del entorno gráfico pueden modificarse para facilitar la lectura del texto en la pantalla?

   Establecer un tamaño de fuente de pantalla grande en la configuración del escritorio hará que todos los textos de la pantalla sean más fáciles de leer.

3. ¿De qué manera puede la aplicación *Orca* ayudar a los usuarios con problemas de visión a interactuar con el entorno de escritorio?

   La aplicación orca puede favorecer a los usuarios con dichos problemas ya que es un lector de pantalla que genera una voz sintetizada para informar de los eventos de la pantalla y para leer el texto bajo el cursor del mouse. También funciona con dispositivos llamados *refreshable braille displays*, para que el usuario pueda identificar el texto con patrones táctiles.



Tema 107: Tareas administrativas



107.1 Administrar cuentas de usuario y de grupo y los archivos de sistema relacionados con ellas

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 107.1

Importancia

5

Áreas de conocimiento clave

- Añadir, modificar y eliminar grupos y usuarios.
- Administrar información de usuarios/grupos en bases de datos de contraseñas/grupos.
- Crear y administrar cuentas limitadas y de propósito especial.

Lista parcial de archivos, términos y utilidades

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`
- `/etc/skel/`
- `chage`
- `getent`
- `groupadd`
- `groupdel`
- `groupmod`
- `passwd`

- `useradd`
- `userdel`
- `usermod`



107.1 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	107 Tareas administrativas
Objetivo:	107.1 Administrar cuentas de usuario y de grupo y los archivos de sistema relacionados con ellas
Lección:	1 de 2

Introducción

La administración de usuarios y grupos es una parte muy importante del trabajo de cualquier administrador de sistemas. Las modernas distribuciones de Linux implementan interfaces gráficas que permiten gestionar todas las actividades relacionadas con este aspecto de forma rápida y sencilla. Estas interfaces son diferentes entre sí en términos de diseños gráficos, no obstante las características siguen siendo las mismas. Con estas herramientas se pueden ver, editar, añadir y eliminar usuarios y grupos locales. Sin embargo, para una gestión más avanzada es necesario trabajar a través de la línea de comandos.

Agregando cuentas de usuario

En Linux, puedes añadir una nueva cuenta de usuario con el comando `useradd`. Por ejemplo, con privilegios de root, puedes crear una nueva cuenta de usuario llamada `Michael` con una configuración por defecto, usando lo siguiente:

```
# useradd michael
```


Cuando se ejecuta el comando `useradd`, la información de usuario y grupo almacenada en las bases de datos de contraseñas y grupos se actualiza para la cuenta de usuario recién creada, y si se especifica, también se crea el directorio principal del nuevo usuario. Adicionalmente se crea un grupo con el mismo nombre de la nueva cuenta de usuario.

Una vez que haya creado el nuevo usuario, puede establecer su contraseña usando el comando `passwd`. Puede revisar su ID de usuario (UID), ID de grupo (GID) y los grupos a los que pertenece a través de los comandos `id` y `groups`.

```
# passwd michael
Changing password for user michael.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
# id michael
uid=1000(michael) gid=100(michael) groups=100(michael)
# groups michael
michael : michael
```

NOTE Recuerde que cualquier usuario puede revisar su UID, GID y los grupos a los que pertenece simplemente usando los comandos `id` y `groups` sin argumentos y que a su vez cualquier usuario puede cambiar su contraseña usando el comando `passwd`. Sin embargo, sólo los usuarios con privilegios de root pueden cambiar la contraseña de *cualquier* usuario.

Las opciones más importantes que se aplican al comando `useradd` son:

`-c`

Crear una nueva cuenta de usuario con comentarios personalizados (por ejemplo, el nombre completo del usuario).

`-d`

Crear una nueva cuenta de usuario con un directorio de inicio específico.

`-e`

Crear una nueva cuenta de usuario estableciendo una fecha específica en la que se desactivará.

`-f`

Crear una nueva cuenta de usuario estableciendo el número de días después de que expire una contraseña, durante los cuales el usuario debe actualizarla (de lo contrario, la cuenta se

desactivará).

`-g`
: Crear una nueva cuenta de usuario con un GID específico.

`-G`
: Crear una nueva cuenta de usuario añadiéndola a múltiples grupos secundarios.

`-k`
: Crear una nueva cuenta de usuario copiando los archivos del "skel" de un directorio personalizado específico (esta opción sólo es válida si se especifica la opción "m" o "crear casa").

`-m`
: Crear una nueva cuenta de usuario con su directorio principal (si no existe).

`-M`
: Crear una nueva cuenta de usuario sin su directorio principal.

`-s`
: Crear una nueva cuenta de usuario con un shell de acceso específico.

`-u`
: Crear una nueva cuenta de usuario con un UID específico.

Vea las páginas de man del comando `useradd` para la lista completa de opciones.

Modificación de las cuentas de usuario

En ocasiones es necesario cambiar un atributo de una cuenta de usuario existente, como también el nombre de usuario, el shell, la fecha de caducidad de la contraseña, etc. En tales casos, necesitas usar el comando `usermod`.

```
# usermod -s /bin/tcsh michael
# usermod -c "Michael User Account" michael
```

Al igual que con el comando `useradd`, el comando `usermod` requiere privilegios de root.

En los ejemplos anteriores, primero se cambia el shell de inicio de sesión de `michael` y luego se añade una breve descripción a esta cuenta de usuario. Recuerda que puedes modificar varios atributos a la vez, especificándolos en un solo comando.

Las opciones más importantes que se aplican al comando `usermod' son:

`-c`

Agrega un breve comentario a la cuenta de usuario.

`-d`

Cambiar el directorio principal de la cuenta de usuario. Cuando se usa con la opción `-m`, los contenidos del directorio principal actual se mueven al nuevo directorio principal, que a su vez se crea de no ser existente.

`-e`

Establece la fecha de expiración de la cuenta de usuario.

`-f`

Establece el número de días después de que una contraseña expira, durante los cuales el usuario debe actualizar la contraseña (de lo contrario la cuenta se desactivará).

`-g`

Cambia el grupo primario de la cuenta de usuario (el grupo debe existir).

`-G`

Añade grupos secundarios a la cuenta de usuario especificada. Cada grupo debe existir y debe estar separado del siguiente por una coma, sin espacios en blanco. Si se usa sola, esta opción elimina todos los grupos existentes a los que el usuario pertenece, mientras que cuando se usa con la opción `-a`, simplemente añade nuevos grupos secundarios a los ya existentes.

`-l`

Cambia el nombre de usuario de la cuenta de usuario especificada.

`-L`

Bloquea la cuenta de usuario especificada. Esto pone un signo de exclamación delante de la contraseña encriptada dentro del archivo `/etc/shadow`, deshabilitando así el acceso con una contraseña para ese usuario.

`-s`

Cambia el shell de acceso de la cuenta de usuario especificada.

`-u`

Cambia el UID de la cuenta de usuario especificada.

`-U`
Desbloquea la cuenta de usuario especificada. Esto elimina el signo de exclamación delante de la contraseña cifrada con el archivo `/etc/shadow`.

Observe las páginas de man del comando `usermod` para la lista completa de opciones.

TIP Recuerde que cuando cambie el nombre de inicio de sesión de una cuenta de usuario, probablemente deba cambiar el nombre del directorio principal de ese usuario junto con otros elementos relacionados con el usuario, por ejemplo, los archivos de la cola de correo. Recuerde también que cuando cambie el UID de una cuenta de usuario, probablemente debería corregir la propiedad de los archivos y directorios fuera del directorio principal del usuario (el ID de usuario se cambia automáticamente para el buzón de correo del usuario y para todos los archivos que son propiedad del usuario y que se encuentran en el directorio principal del usuario).

Eliminando cuentas de usuario

Si quieres borrar una cuenta de usuario, puedes usar el comando `userdel`. En particular, este comando actualiza la información almacenada en las bases de datos de las cuentas, borrando todas las entradas referentes al usuario especificado. La opción `-r` también elimina el directorio principal del usuario y todos sus contenidos, junto con el spool de correo del usuario. Otros archivos, localizados en otros lugares, deben ser buscados y eliminados manualmente.

```
# userdel -r michael
```

En cuanto a `useradd` y `usermod`, necesitas el privilegio de root para borrar cuentas de usuario. === Agregando, modificando y eliminando grupos

Al igual que con la gestión de usuarios, puede añadir, modificar y eliminar grupos usando los comandos `groupadd`, `groupmod` y `groupdel` con privilegios de root. Si quiere crear un nuevo grupo llamado `developer`, puede ejecutar el siguiente comando:

```
# groupadd -g 1090 developer
```

La opción `-g` de este comando crea un grupo con un GID específico.

WARNING Recuerde que cuando añade una nueva cuenta de usuario, el grupo primario y los grupos secundarios a los que pertenece *deben* existir antes de lanzar el comando `useradd`.

Si luego deseas renombrar el grupo de `developer` a `web-developer` y cambiar su GID, puedes ejecutar lo siguiente:

```
# groupmod -n web-developer -g 1050 developer
```

TIP Recuerde que si cambia el GID usando la opción `-g`, debe cambiar el GID de todos los archivos y directorios que deben seguir perteneciendo al grupo.

Finalmente, si quieres borrar el grupo de `web-developer`, puedes ejecutar lo siguiente:

```
# groupdel web-developer
```

No se puede eliminar un grupo si es el grupo principal de una cuenta de usuario. Por lo tanto, debe eliminar el usuario antes de eliminar el grupo. En cuanto a los usuarios, si elimina un grupo, los archivos pertenecientes a ese grupo permanecen en su sistema de archivos y no se eliminan ni se asignan a otro grupo.

El directorio skel

Cuando añades una nueva cuenta de usuario, incluso creando su directorio principal, el directorio principal recién creado se carga de archivos y carpetas que se copian del directorio skel (por defecto `/etc/skel`). La idea detrás de esto es simple: un administrador del sistema quiere agregar nuevos usuarios que tengan los mismos archivos y directorios en su carpeta principal. Por lo tanto, si desea personalizar los archivos y carpetas que se crean automáticamente en el directorio principal, debe añadir estos nuevos archivos y carpetas al directorio skel.

TIP Recuerde que si desea listar todos los archivos y directorios en el directorio skel, debe usar el comando `ls -al`.

El archivo `/etc/login.defs`

En Linux, el archivo `/etc/login.defs` especifica los parámetros de configuración que controlan la creación de usuarios y grupos. Además, los comandos mostrados en las secciones anteriores toman por defecto valores de este archivo.

Las directivas más importantes son:

`UID_MIN` y `UID_MAX`
: El rango de ID de usuario que puede ser asignado a los nuevos usuarios ordinarios.

`GID_MIN` y `GID_MAX`
: El rango de ID de grupo que puede ser asignado a nuevos grupos ordinarios.

`CREATE_HOME`
: Especifica si un directorio principal debe ser creado por defecto para los nuevos usuarios.

`USERGROUPS_ENAB`
: Especifica si el sistema debe crear por defecto un nuevo grupo para cada nueva cuenta de usuario con su mismo nombre, y a su vez al eliminar la cuenta también se debe eliminar el grupo principal del usuario si ya no contiene miembros.

`MAIL_DIR`
: El directorio de la cola de correo.

`PASS_MAX_DAYS`
: El número máximo de días que una contraseña puede ser usada.

`PASS_MIN_DAYS`
: El número mínimo de días permitido entre los cambios de contraseña.

`PASS_MIN_LEN`
: La longitud mínima aceptable de la contraseña.

`PASS_WARN_AGE`
: El número de días de advertencia antes de que una contraseña expire.

> **TIP** Cuando administre usuarios y grupos, compruebe siempre este archivo para ver y eventualmente cambiar el comportamiento por defecto del sistema (si fuese necesario).

El comando `passwd`

Este comando se utiliza principalmente para cambiar la contraseña de un usuario. Como se describió anteriormente, cualquier usuario puede cambiar su propia contraseña, pero sólo root puede cambiar

la contraseña de *cualquier* usuario. Esto sucede porque el comando `passwd` tiene el bit SUID puesto (una `s` en el lugar del flag ejecutable para el propietario), lo que significa que se ejecuta con los privilegios del propietario del archivo (por lo tanto, root).

```
# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 42096 mag 17  2015 /usr/bin/passwd
```

Dependiendo de las opciones de `passwd` utilizadas, puedes controlar aspectos específicos del envejecimiento de las contraseñas tales como:

`-d`
: Borrar la contraseña de una cuenta de usuario (deshabilitando así al usuario).

`-e`
: Forzar a la cuenta de usuario a cambiar la contraseña.

`-i`
: Establecer el número de días de inactividad después de que una contraseña expire, durante los cuales el usuario debe actualizar la contraseña (de lo contrario, la cuenta será desactivada).

`-l`
: Bloquea la cuenta de usuario (la contraseña cifrada se prefija con un signo de exclamación en el archivo `/etc/shadow`).

`-n`
: Establece la duración mínima de la contraseña.

`-S`
: Información de salida sobre el estado de la contraseña de una cuenta de usuario específica.

`-u`
: Desbloquea la cuenta de usuario (el signo de exclamación se elimina del campo de la contraseña en el archivo `/etc/shadow`).

`-x`
: Establece la duración máxima de la contraseña.

`-w`
: Determina el número de días de advertencia antes de que la contraseña expire, durante los cuales

se advierte al usuario que debe cambiarla.

> **NOTE**
>
> Los grupos también pueden tener una contraseña, que puede ser establecida usando el comando `gpasswd`. Los usuarios que no son miembros de un grupo pero conocen su contraseña, pueden unirse a este temporalmente usando el comando `newgrp`. Recuerde que `gpasswd` también se utiliza para añadir y eliminar usuarios de un grupo y para establecer la lista de administradores y miembros ordinarios del grupo.

El comando `chage`

Este comando determinado como "change age", se usa para cambiar el período de la contraseña de un usuario. El comando `chage` está restringido a root, excepto la opción `-l`, que puede ser usada por usuarios ordinarios para listar el tiempo de su contraseña.

Las otras opciones que se aplican al comando `chage` son:

`-d`

Establece el último cambio de contraseña para una cuenta de usuario.

`-E`

Establece la fecha de caducidad de una cuenta de usuario.

`-I`

Establece el número de días de inactividad después de que una contraseña expira, durante los cuales el usuario deberá actualizarla (de lo contrario la cuenta será desactivada).

`-m`

Establece la duración mínima de la contraseña para una cuenta de usuario.

`-M`

Establece la duración máxima de la contraseña para una cuenta de usuario.

`-W`

Establece el número de días de advertencia antes que la contraseña expire, durante los cuales se le advierte al usuario que deberá cambiarla.

Ejercicios guiados

1. Para cada uno de los siguientes comandos, identifique el propósito correspondiente:

`usermod -L`	
`passwd -u`	
`chage -E`	
`groupdel`	
`useradd -s`	
`groupadd -g`	
`userdel -r`	
`usermod -l`	
`groupmod -n`	
`useradd -m`	

2. Para cada uno de los siguientes comandos `passwd`, identifique el correspondiente comando `chage`:

`passwd -n`	
`passwd -x`	
`passwd -w`	
`passwd -i`	
`passwd -S`	

3. Explique en detalle el propósito de los comandos en la pregunta anterior:

4. ¿Qué comandos puedes usar para bloquear una cuenta de usuario? ¿Y qué comandos para desbloquearla?

Ejercicios de exploración

1. Usando el comando `groupadd`, crea los grupos de `administrators` y `developers`. Asume que estás trabajando como root.

2. Ahora que ha creado estos grupos, ejecute el siguiente comando: `useradd -G administrators,developers kevin`. ¿Qué operaciones realiza este comando? Asume que `CREATE_HOME` y `USERGROUPS_ENAB` están configurados en `yes` en el archivo `/etc/login.defs`.

3. Crea un nuevo grupo llamado `designers`, renómbrelo a `web-designers` y añada este nuevo grupo a los grupos secundarios de la cuenta de usuario `kevin`. Identifica todos los grupos a los que pertenece `kevin` y sus identificaciones.

4. Elimina sólo el grupo de `developers` de los grupos secundarios de `kevin`.

5. Establezca la contraseña de la cuenta de usuario `kevin`.

6. Usando el comando `chage`, primero compruebe la fecha de caducidad de la cuenta de usuario `kevin` y luego cámbiela al 31 de diciembre de 2022. ¿Qué otro comando puedes usar para cambiar la fecha de caducidad de una cuenta de usuario?

7. Añade una nueva cuenta de usuario llamada `emma` con el UID 1050 y establezca `administrators` como su grupo primario y `developers` y `web-designers` como sus grupos secundarios.

8. Cambie el shell de inicio de sesión de `emma` a `/bin/sh`.

9. Elimina las cuentas de usuario `emma` y `kevin` y los grupos de `administrators`, `developers` y `web-designers`

Resumen

En esta lección aprendió:

- Los fundamentos de la gestión de usuarios y grupos en Linux.
- Agregar, modificar y eliminar cuentas de usuario.
- Agregar, modificar y eliminar cuentas de grupo.
- Mantener el directorio del esqueleto.
- Editar el archivo que controla la creación de usuarios y grupos.
- Cambiar las contraseñas de las cuentas de usuario.
- Cambiar la información sobre la antigüedad de las contraseñas de las cuentas de usuario.

Los siguientes archivos y comandos fueron discutidos en esta lección:

`useradd`
: Crear una nueva cuenta de usuario.

`usermod`
: Modificar una cuenta de usuario.

`userdel`
: Eliminar una cuenta de usuario.

`groupadd`
: Crear una nueva cuenta de grupo.

`groupmod`
: Modificar una cuenta de grupo.

`groupdel`
: Eliminar una cuenta de grupo.

`passwd`
: Cambiar la contraseña de las cuentas de usuario y controlar todos los aspectos del envejecimiento de la contraseña.

`chage`
: Cambiar la información de caducidad de la contraseña del usuario.

`/etc/skel`
: La ubicación por defecto del directorio skel.

`/etc/login.defs`
: El archivo que controla la creación de usuarios y grupos proporciona valores por defecto para varios parámetros de la cuenta de usuario.

Respuesta a los ejercicios guiados

1. Para cada uno de los siguientes comandos, identifique el propósito correspondiente:

`usermod -L`	Bloquea una cuenta de usuario.
`passwd -u`	Desbloquea una cuenta de usuario.
`chage -E`	Establece la fecha de caducidad de una cuenta de usuario.
`groupdel`	Elimina un grupo.
`useradd -s`	Crea una nueva cuenta de usuario con un nombre de usuario específico.
`groupadd -g`	Crea un nuevo grupo con un GID específico.
`userdel -r`	Elimina una cuenta de usuario, su directorio principal, su contenido y el spool de correo del usuario.
`usermod -l`	Cambia el nombre de usuario de una cuenta.
`groupmod -n`	Cambia el nombre del grupo.
`useradd -m`	Crea una nueva cuenta de usuario y su directorio de inicio

2. Para cada uno de los siguientes comandos `passwd`, identifique el comando correspondiente `chage`:

`passwd -n`	`chage -m`
`passwd -x`	`chage -M`
`passwd -w`	`chage -W`
`passwd -i`	`chage -I`
`passwd -S`	`chage -l`

3. Explique en detalle el propósito de los comandos en la pregunta anterior:

   En Linux, puede usar el comando `passwd -n` (o `chage -m`) para establecer el número mínimo de días entre cambios de contraseña, el comando `passwd -x` (o `chage -M`) para establecer el número máximo de días durante los cuales una contraseña es válida, el comando `passwd -w` (o `chage -W`) para establecer el número de días de advertencia antes de que la contraseña expire, el comando `passwd -i` (o `chage -I`) para establecer el número de días de inactividad durante los cuales el

usuario debe cambiar la contraseña y el comando `passwd -S` (o `chage -l`) para mostrar una breve información sobre la contraseña de la cuenta de usuario.

4. ¿Qué comandos puedes usar para bloquear una cuenta de usuario? ¿Y qué comandos para desbloquearla?

   Si quieres bloquear una cuenta de usuario, puedes usar uno de estos comandos: `usermod -L`, `usermod --lock` y `passwd -l`. En cambio, si quieres desbloquearla, puedes usar `usermod -U`, `usermod --unlock` y `passwd -u`.

Respuestas a los ejercicios de exploración

1. Usando el comando `groupadd`, crea los grupos de `administrators` y `developers`. Asume que estás trabajando como root.

   ```
   # groupadd administrators
   # groupadd developers
   ```

2. Ahora que ha creado estos grupos, ejecute el siguiente comando: `useradd -G administrators,developers kevin`. ¿Qué operaciones realiza este comando? Asume que `CREATE_HOME` y `USERGROUPS_ENAB` están configurados en `yes` en el archivo `/etc/login.defs`.

   El comando añade un nuevo usuario, llamado `kevin` a la lista de usuarios del sistema, crea su directorio principal (`CREATE_HOME` está establecido en `yes` y por lo tanto puedes omitir la opción -m) y crea un nuevo grupo, llamado `kevin`, como el grupo principal de esta cuenta de usuario (`USERGROUPS_ENAB` está establecido en `yes`). Finalmente, los archivos y carpetas dentro del directorio skel se copian al directorio principal de `kevin`.

3. Crea un nuevo grupo llamado `designers`, renómbrelo a `web-designers` y añada este nuevo grupo a los grupos secundarios de la cuenta de usuario `kevin`. Identifique todos los grupos a los que pertenece `kevin` y sus identificaciones.

   ```
   # groupadd designers
   # groupmod -n web-designers designers
   # usermod -a -G web-designers kevin
   # id kevin
   uid=1010(kevin) gid=1030(kevin)
   groups=1030(kevin),1028(administrators),1029(developers),1031(web-designers)
   ```

4. Elimine sólo el grupo de "developers" de los grupos secundarios de `kevin`.<<<<<

   ```
   # usermod -G administrators,web-designers kevin
   # id kevin
   uid=1010(kevin) gid=1030(kevin)
   groups=1030(kevin),1028(administrators),1031(web-designers)
   ```

   El comando `usermod` no tiene la opción de eliminar sólo un grupo; por lo tanto, es necesario especificar todos los grupos secundarios a los que pertenece el usuario.

5. Establezca la contraseña de la cuenta de usuario `kevin`.

```
# passwd kevin
Changing password for user kevin.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

6. Usando el comando `chage`, primero compruebe la fecha de caducidad de la cuenta de usuario `kevin` y luego cámbiela al 31 de diciembre de 2022. ¿Qué otro comando puedes usar para cambiar la fecha de caducidad de una cuenta de usuario?

```
# chage -l kevin | grep "Account expires"
Account expires     : never
# chage -E 2022-12-31 kevin
# chage -l kevin | grep "Account expires"
Account expires     : dec 31, 2022
```

El comando `usermod` con la opción `-e` es equivalente a `chage -E`.

7. Añade una nueva cuenta de usuario llamada `emma` con el UID 1050 y establece `administrators` como su grupo primario y `developers` y `web-designers` como sus grupos secundarios.

```
# useradd -u 1050 -g administrators -G developers,web-designers emma
# id emma
uid=1050(emma) gid=1028(administrators)
groups=1028(administrators),1029(developers),1031(web-designers)
```

8. Cambie el shell de inicio de sesión de `emma` a `/bin/sh`.

```
# usermod -s /bin/sh emma
```

9. Elimina las cuentas de usuario `emma` y `kevin` y los grupos de `administrators`, `developers` y `web-designers`

```
# userdel -r emma
# userdel -r kevin
# groupdel administrators
# groupdel developers
# groupdel web-designers
```



107.1 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	107 Tareas administrativas
Objetivo:	107.1 Administrar cuentas de usuario y de grupo y los archivos de sistema relacionados con ellas
Lección:	1 de 2

Introducción

Las herramientas de línea de comando discutidas en la lección anterior y las aplicaciones gráficas proporcionadas por cada distribución que realizan las mismas tareas, actualizan una serie de archivos que almacenan información sobre los usuarios y los grupos.

Estos archivos se encuentran en el directorio `/etc/` y son los siguientes:

`/etc/passwd`
: Un archivo de siete campos delimitados por dos puntos que contienen información básica sobre los usuarios.

`/etc/group`
: Un archivo de cuatro campos delimitados por dos puntos que contienen información básica sobre los grupos.

`/etc/shadow`
: Un archivo de nueve campos delimitados por dos puntos que contienen contraseñas encriptadas

de usuario.

`/etc/gshadow`
Un archivo de cuatro campos delimitados por dos puntos que contienen contraseñas de grupo encriptadas.

Aunque estos cuatro archivos están en texto plano, no deben ser editados directamente, sino siempre a través de las herramientas que proporciona la distribución que está utilizando.

/etc/passwd

Este es un archivo legible por el mundo que contiene una lista de usuarios, cada uno en una línea separada. Cada línea consiste en siete campos delimitados por dos puntos:

Nombre de usuario
El nombre utilizado cuando el usuario se conecta al sistema.

Contraseña
Se encuentra encriptada (o una `x` si se usa el archivo /etc/shadow).

ID de usuario (UID)
El número de identificación asignado al usuario en el sistema.

ID de grupo (GID)
El número de grupo primario del usuario en el sistema.

GECOS
Un campo de comentario opcional, que se utiliza para añadir información adicional sobre el usuario (como el nombre completo). El campo puede contener múltiples entradas separadas por comas.

Home Directory
La ruta absoluta del directorio principal del usuario.

Shell
La ruta absoluta del programa que abre cuando el usuario se conecta al sistema (normalmente un shell interactivo como `/bin/bash`).

/etc/group

Este es un archivo legible por el mundo que contiene una lista de grupos, cada uno en una línea separada. Cada línea consiste en cuatro campos delimitados por dos puntos:

Nombre de grupo
El nombre de grupo.

Contraseña de grupo
La contraseña cifrada del grupo (o una `x` si se usa el archivo /etc/gshadow).

ID de grupo (GID)
El número de identificación asignado al grupo en el sistema.

Lista de miembros
Una lista delimitada por comas de los usuarios que pertenecen al grupo, excepto aquellos para los que esten en el grupo principal.

/etc/shadow

Se trata de un archivo legible sólo por root y por usuarios con privilegios de root que contiene contraseñas de usuario cifradas, cada una en una línea separada. Cada línea consiste en nueve campos delimitados por dos puntos:

Nombre de usuario
El nombre usado cuando el usuario se conecta al sistema.

Contraseña encriptada
La contraseña encriptada del usuario (si el valor comienza con `!`, la cuenta está bloqueada).

Fecha del último cambio de contraseña
La fecha del último cambio de contraseña, como número de días desde el 01/01/1970 (un valor de 0 significa que el usuario debe cambiar la contraseña la próxima vez que se conecte).

Edad mínima de la contraseña
El número mínimo de días después de un cambio de contraseña, que debe pasar antes de que el usuario pueda cambiar la contraseña de nuevo.

Edad máxima de la contraseña
El número máximo de días que debe pasar antes de que se requiera un cambio de contraseña.

Período de alerta de la contraseña

El número de días, antes de que la contraseña expire, durante los cuales se advierte al usuario que debe cambiarla.

Periodo de inactividad de la contraseña

El número de días después de que una contraseña expira, durante los cuales el usuario debe actualizarla. Después de este período, si el usuario no cambia la contraseña, la cuenta se desactivará.

Fecha de vencimiento de la cuenta

La fecha expresada como el número de días desde el 01/01/1970, en que la cuenta de usuario será desactivada (un campo vacío significa que la cuenta de usuario nunca expirará).

Un campo reservado

Un campo para un uso futuro.

/etc/gshadow

Se trata de un archivo legible sólo por root y por usuarios con privilegios de root que contiene contraseñas de grupo cifradas, cada una en una línea separada. Cada línea consiste en cuatro campos delimitados por dos puntos:

Nombre del grupo

El nombre del grupo.

Contraseña encriptada

La contraseña encriptada del grupo (se usa cuando un usuario, que no es miembro del grupo, quiere unirse al grupo usando el comando `newgrp` — si la contraseña comienza con `!`, nadie puede acceder al grupo con `newgrp`).

Administradores del Grupo

Una lista delimitada por comas de los administradores del grupo. Ellos pueden cambiar la contraseña del grupo y pueden agregar o quitar miembros del grupo con el comando `gpasswd`.

Miembros del grupo

Una lista delimitada por comas de los miembros del grupo.

Filtrar las bases de datos de contraseñas y grupos

Muy a menudo puede ser necesario revisar la información sobre los usuarios y grupos almacenados

en estos cuatro archivos así como también buscar registros específicos. Para realizar esta tarea, puede utilizar el comando `grep` o alternativamente concatenar `cat` y `grep`.

```
# grep emma /etc/passwd
emma:x:1020:1020:User Emma:/home/emma:/bin/bash
# cat /etc/group | grep db-admin
db-admin:x:1050:grace,frank
```

Otra forma de acceder a estas bases de datos es usar el comando `getent`. En general, este comando muestra las entradas de las bases de datos soportadas por las bibliotecas *Name Service Switch* (NSS) y requiere el nombre de la base de datos y una clave de búsqueda. Si no se proporciona ningún argumento de clave, se muestran todas las entradas de la base de datos especificada (a menos que la base de datos no soporte la enumeración). De lo contrario, si se proporcionan uno o más argumentos clave, la base de datos se filtra en consecuencia.

```
# getent passwd emma
emma:x:1020:1020:User Emma:/home/emma:/bin/bash
# getent group db-admin
db-admin:x:1050:grace,frank
```

El comando `getent` no requiere la autoridad de la raíz; sólo necesitas poder leer la base de datos de la que quieres recuperar los registros.

NOTE Recuerda que `getent` sólo puede acceder a las bases de datos configuradas en el archivo `/etc/nsswitch.conf`.

Ejercicios guiados

1. Observe la siguiente salida y responda a las siguientes preguntas:

```
# cat /etc/passwd | grep '\(root\|mail\|catherine\|kevin\)'
root:x:0:0:root:/root:/bin/bash
mail:x:8:8:mail:/var/spool/mail:/sbin/nologin
catherine:x:1030:1025:User Chaterine:/home/catherine:/bin/bash
kevin:x:1040:1015:User Kevin:/home/kevin:/bin/bash
# cat /etc/group | grep '\(root\|mail\|db-admin\|app-developer\)'
root:x:0:
mail:x:8:
db-admin:x:1015:emma,grace
app-developer:x:1016:catherine,dave,christian
# cat /etc/shadow | grep '\(root\|mail\|catherine\|kevin\)'
root:$6$1u36Ipok$ljt8ooPMLewAhkQPf.lYgGopAB.jClTO6ljsdczxvkLPkpi/amgp.zyfAN680z
rLLp2avvpdKA0llpssdfcPppOp:18015:0:99999:7:::
mail:*:18015:0:99999:7:::
catherine:$6$ABCD25jlld14hpPthEFGnnssEWw1234yioMpliABCdef1f3478kAfhhAfgbAMjY1/B
AeeAsl/FeEdddKd12345g6kPACcik:18015:20:90:5:::
kevin:$6$DEFGabc123WrLp223fsvp0ddx3dbA7pPPc4LMaa123u6Lp02Lpvm123456pyphhh5ps012
vbArL245.PR1345kkA3Gas12P:18015:0:60:7:2::
# cat /etc/gshadow | grep '\(root\|mail\|db-admin\|app-developer\)'
root:*::
mail:*::
db-admin:!:emma:emma,grace
app-developer:!::catherine,dave,christian
```

- ¿Cuál es el ID de usuario (UID) y el ID de grupo (GID) de `root` y `catherine` ?

- ¿Cuál es el nombre del grupo primario de `Kevin`? ¿Hay otros miembros en este grupo?

- ¿Cuál shell está asignado para el `mail`? ¿Qué significa?

- ¿Quiénes son los miembros del grupo de `app-developer`? ¿Cuáles de estos miembros son los administradores del grupo y cuáles son los miembros ordinarios?

- ¿Cuál es la duración mínima de la contraseña para `catherine`? ¿Y cuál es la duración máxima de la contraseña?

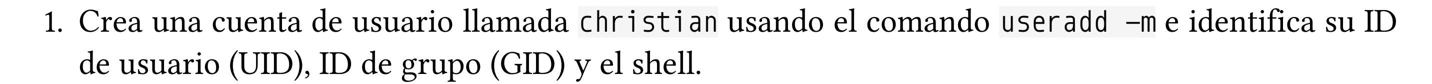
- ¿Cuál es el período de inactividad de la contraseña para `kevin`?

2. Por convención, ¿Qué identificaciones se asignan a las cuentas del sistema y cuáles a los usuarios ordinarios?

3. ¿Cómo puede saber si una cuenta de usuario, que antes podía acceder al sistema, ahora se encuentra bloqueada? Supongamos que su sistema utiliza contraseñas en la sombra.

Ejercicios de exploración

1. Crea una cuenta de usuario llamada `christian` usando el comando `useradd -m` e identifica su ID de usuario (UID), ID de grupo (GID) y el shell.



2. Identifica el nombre del grupo primario de `christian`. ¿Qué puedes deducir?

3. Usando el comando `getent`, revisa la información de la contraseña de la cuenta del usuario `christian`.

4. Añade el grupo `editor` a los grupos secundarios de `christian`. Supongamos que este grupo ya contiene a `Emma`, `Dave` y `Frank` como miembros ordinarios. ¿Cómo puedes verificar que no hay administradores para este grupo?

5. Ejecute el comando `ls -l /etc/passwd /etc/group /etc/shadow /etc/gshadow` y describa la salida que imprime en términos de permisos de archivo. ¿Cuál de estos cuatro archivos están con "shadow" por razones de seguridad? Supongamos que tu sistema utiliza contraseñas shadow.

Resumen

En esta lección aprendió:

- La ubicación de los archivos que almacenan información sobre usuarios y grupos.
- Administrar la información de usuarios y grupos almacenada en bases de datos de contraseñas y grupos.
- Recuperar información de las bases de datos de contraseñas y grupos.

Los siguientes archivos y comandos fueron discutidos en esta lección:

`/etc/passwd`
El archivo que contiene información básica sobre los usuarios.

`/etc/group`
El archivo que contiene información básica sobre los grupos.

`/etc/shadow`
El archivo que contiene las contraseñas encriptadas de los usuarios.

`/etc/gshadow`
El archivo que contiene las contraseñas encriptadas de un grupo .

`getent`
Filtrar las bases de datos de contraseñas y grupos.

Respuesta a los ejercicios guiados

1. Observe la siguiente salida y responda a las siguientes preguntas:

```
# cat /etc/passwd | grep '\(root\|mail\|catherine\|kevin\)'
root:x:0:0:root:/root:/bin/bash
mail:x:8:8:mail:/var/spool/mail:/sbin/nologin
catherine:x:1030:1025:User Chaterine:/home/catherine:/bin/bash
kevin:x:1040:1015:User Kevin:/home/kevin:/bin/bash
# cat /etc/group | grep '\(root\|mail\|db-admin\|app-developer\)'
root:x:0:
mail:x:8:
db-admin:x:1015:emma,grace
app-developer:x:1016:catherine,dave,christian
# cat /etc/shadow | grep '\(root\|mail\|catherine\|kevin\)'
root:$6$1u36Ipok$ljt8ooPMLewAhkQPf.lYgGopAB.jClTO6ljsdczxvkLPkpi/amgp.zyfAN680z
rLLp2avvpdKA0llpssdfcPppOp:18015:0:99999:7:::
mail:*:18015:0:99999:7:::
catherine:$6$ABCD25jlld14hpPthEFGnnssEWw1234yioMpliABCdef1f3478kAfhhAfgbAMjY1/B
AeeAsl/FeEdddKd12345g6kPACcik:18015:20:90:5:::
kevin:$6$DEFGabc123WrLp223fsvp0ddx3dbA7pPPc4LMaa123u6Lp02Lpvm123456pyphhh5ps012
vbArL245.PR1345kkA3Gas12P:18015:0:60:7:2::
# cat /etc/gshadow | grep '\(root\|mail\|db-admin\|app-developer\)'
root:*::
mail:*::
db-admin:!:emma:emma,grace
app-developer:!::catherine,dave,christian
```

- ¿Cuál es el ID de usuario (UID) y el ID de grupo (GID) de `root` y `catherine` ?

  El UID y GID de `root` son 0 y 0, mientras que el UID y GID de `catherine` son 1030 y 1025.

- ¿Cuál es el nombre del grupo primario de `Kevin`? ¿Hay otros miembros en este grupo?

  El nombre del grupo es `db-admin`. También `emma` y `grace` están en este grupo.

- ¿Cuál shell está asignado para el `mail`? ¿Qué significa?

  `mail` es una cuenta de usuario del sistema y su shell es `/sbin/nologin`. De hecho, las cuentas de usuario del sistema como `mail`, `ftp`, `news` y `daemon` se utilizan para realizar tareas administrativas y por lo tanto se debe evitar el inicio de sesión normal para estas cuentas. Por eso el shell está normalmente configurado como `/sbin/nologin` o `/bin/false`.

- ¿Quiénes son los miembros del grupo de `app-developer`? ¿Cuáles de estos miembros son los administradores del grupo y cuáles son los miembros ordinarios?

  Los miembros son `catherine`, `dave` y `christian`. Todos son miembros ordinarios.

- ¿Cuál es la duración mínima de la contraseña para `catherine`? ¿Y cuál es la duración máxima de la contraseña?

  La duración mínima de la contraseña es de 20 días, mientras que la duración máxima es de 90 días.

- ¿Cuál es el período de inactividad de la contraseña para `kevin`?

  El período de inactividad de la contraseña es de 2 días. Durante este período `kevin` debe actualizar la contraseña, de lo contrario la cuenta será desactivada.

2. Por convención, ¿Qué identificaciones se asignan a las cuentas del sistema y cuáles a los usuarios ordinarios?

   Las cuentas del sistema suelen tener UIDs inferiores a 100 o entre 500 y 1000, mientras que los usuarios ordinarios tienen UIDs a partir de 1000, aunque algunos sistemas heredados pueden empezar a numerar a 500. El usuario `root` tiene UID 0. Recuerde que los valores `UID_MIN` y `UID_MAX` en `/etc/login.defs` definen el rango de UIDs usados para la creación de usuarios ordinarios. Desde el punto de vista de LPI Linux Essentials y LPIC-1, las cuentas del sistema tienen UIDs menores que 1000 y los usuarios ordinarios tienen UIDs mayores que 1000.

3. ¿Cómo puede saber si una cuenta de usuario, que antes podía acceder al sistema, ahora se encuentra bloqueada? Supongamos que su sistema utiliza contraseñas en la sombra.

   Cuando se utilizan contraseñas en la sombra, el segundo campo de `/etc/passwd` contiene el carácter `x` para cada cuenta de usuario, porque las contraseñas cifradas de los usuarios se almacenan en `/etc/shadow`. En concreto, la contraseña encriptada de una cuenta de usuario se almacena en el segundo campo de este archivo y, si comienza con un signo de exclamación, la cuenta está bloqueada.

Respuestas a los ejercicios de exploración

1. Crea una cuenta de usuario llamada `christian` usando el comando `useradd -m` e identifica su ID de usuario (UID), ID de grupo (GID) y el shell.

```
# useradd -m christian
# cat /etc/passwd | grep christian
christian:x:1050:1060::/home/christian:/bin/bash
```

El UID y el GID de `christian` son 1050 y 1060 respectivamente (el tercer y cuarto campo en `/etc/passwd`). `/bin/bash` es el shell de esta cuenta de usuario (el séptimo campo en `/etc/passwd`).

2. Identifica el nombre del grupo primario de `christian`. ¿Qué puedes deducir?

```
# cat /etc/group | grep 1060
christian:x:1060:
```

El nombre del grupo primario de `christian` es `christian` (el primer campo en `/etc/group`). Por lo tanto, `USERGROUPS_ENAB` en `/etc/login.defs` está configurado para que `useradd` por defecto cree un grupo con el mismo nombre de la cuenta de usuario.

3. Usando el comando `getent`, revisa la información de la contraseña de la cuenta del usuario `christian`.

```
# getent shadow christian
christian:!:18015:0:99999:7:::
```

La cuenta de usuario `christian` no tiene la contraseña establecida y ahora está bloqueada (el segundo campo en `/etc/shadow` contiene un signo de exclamación). No hay una edad mínima y máxima para la contraseña de este usuario (el cuarto y quinto campo están fijados en 0 y 99999 días), mientras que el período de advertencia de la contraseña está fijado en 7 días (el sexto campo). Finalmente, no hay período de inactividad (el séptimo campo) y la cuenta nunca expira (el octavo campo).

4. Añade el grupo `editor` a los grupos secundarios de `christian`. Supongamos que este grupo ya contiene a `Emma`, `Dave` y `Frank` como miembros ordinarios. ¿Cómo puedes verificar que no hay administradores para este grupo?

```
# cat /etc/group | grep editor
editor:x:1100:emma,dave,frank
# usermod -a -G editor christian
# cat /etc/group | grep editor
editor:x:1100:emma,dave,frank,christian
# cat /etc/gshadow | grep editor
editor:!::emma,dave,frank,christian
```

El tercer y cuarto campo en `/etc/ghadow` contienen los administradores y miembros ordinarios del grupo especifico. Por lo tanto, como el tercer campo está vacío para `editor`, no hay administradores para este grupo (`emma`, `dave`, `frank` y `christian` son todos miembros ordinarios).

5. Ejecute el comando `ls -l /etc/passwd /etc/group /etc/shadow /etc/gshadow` y describa la salida que imprime en términos de permisos de archivo. ¿Cuál de estos cuatro archivos están con "shadow" por razones de seguridad? Supongamos que tu sistema utiliza contraseñas shadow.

```
# ls -l /etc/passwd /etc/group /etc/shadow /etc/gshadow
-rw-r--r-- 1 root root    853 mag  1 08:00 /etc/group
-rw-r----- 1 root shadow 1203 mag  1 08:00 /etc/gshadow
-rw-r--r-- 1 root root   1354 mag  1 08:00 /etc/passwd
-rw-r----- 1 root shadow 1563 mag  1 08:00 /etc/shadow
```

Los archivos `/etc/passwd` y `/etc/group` son legibles en todo el mundo y están ocultos por razones de seguridad. Cuando se usan las contraseñas shadow, puedes observar una `x` en el segundo campo de estos archivos, porque las contraseñas encriptadas de los usuarios y grupos se almacenan en `/etc/shadow` y `/etc/gshadow` respectivamente, que son legibles sólo por root. Incluso por los miembros pertenecientes al grupo `shadow`.



107.2 Automatizar tareas administrativas del sistema mediante la programación de trabajos

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 107.2

Importancia

4

Áreas de conocimiento clave

- Gestionar trabajos con cron y con at.
- Configurar el acceso del usuario a los servicios de cron y at.
- Entender las unidades de temporización de systemd.

Lista parcial de archivos, términos y utilidades

- `/etc/cron.{d,daily,hourly,monthly,weekly}/`
- `/etc/at.deny`
- `/etc/at.allow`
- `/etc/crontab`
- `/etc/cron.allow`
- `/etc/cron.deny`
- `/var/spool/cron/`
- `crontab`
- `at`
- `atq`

- `atrm`
- `systemctl`
- `systemd-run`



107.2 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	107 Tareas administrativas
Objetivo:	107.2 Automatizar tareas administrativas del sistema mediante la programación de trabajos
Lección:	1 de 2

Introducción

Automatizar las tareas regulares del sistema mediante la programación de trabajos (o Jobs) es una de las cosas más importantes que un buen administrador debe hacer. Por ejemplo, un administrador puede crear y automatizar trabajos para copias de seguridad, actualizaciones del sistema y para realizar muchas otras actividades repetitivas. Para ello puede utilizar la función `cron`, que es útil para automatizar la programación periódica de trabajos.

Programar trabajos con Cron

En sistemas Linux, `cron` es un demonio que se ejecuta continuamente y se activa cada minuto para comprobar un conjunto de tablas en busca de tareas a ejecutar. Estas tablas se conocen como *crontabs* y contienen las llamadas *cron jobs*. Cron es adecuado para servidores y sistemas que están encendidos constantemente, porque cada trabajo de cron se ejecuta sólo si el sistema se está ejecutando a la hora programada. Puede ser utilizado por usuarios ordinarios, cada uno de los cuales tiene su propio `crontab`, así como el usuario root que gestiona los crontabs del sistema.

> **NOTE** En Linux también existe la utilidad `anacron`, que es adecuada para sistemas que pueden ser apagados (como computadoras de escritorio o portátiles). Este sólo puede ser usado por root. Si la máquina está apagada cuando los trabajos de `anacron` deben ser ejecutados, se ejecutarán la próxima vez que se encienda la máquina. `Anacron` está fuera del alcance de la certificación LPIC-1.

Crontabs de usuario

Los crontabs de usuario son archivos de texto que gestionan la programación de los trabajos cron definidos por el usuario. Siempre tienen el nombre de la cuenta de usuario que los creó, pero la ubicación de estos archivos depende de la distribución utilizada (generalmente un subdirectorio de `/var/spool/cron`).

Cada línea en un crontab de usuario contiene seis campos separados por un espacio:

- El minuto de la hora (0-59).
- La hora del día (0-23).
- El día del mes (1-31).
- El mes del año (1-12).
- El día de la semana (0-7 con domingo=0 o domingo=7).
- La orden a ejecutar.

Para el mes del año y el día de la semana puede usar las tres primeras letras del nombre en lugar del número correspondiente.

Los primeros cinco campos indican cuándo ejecutar el comando que se especifica en el sexto campo, y pueden contener uno o más valores. En particular, se pueden especificar múltiples valores utilizando:

`*` (asterisco)
: Se refiere a cualquier valor.

`,` (coma)
: Especifica una lista de posibles valores.

`-` (guión)
: Especifica un rango de valores posibles.

/ (slash)
Especifica valores escalonados.

Muchas distribuciones incluyen el archivo `/etc/crontab` que puede ser usado como referencia para la disposición de un archivo `cron`. A continuación se muestra un ejemplo de archivo `/etc/crontab` de una instalación de Debian:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR
sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
```

Crontabs de sistema

Los crontabs de sistema son archivos de texto que gestionan la programación de los trabajos del cron del sistema y sólo pueden ser editados por el usuario root. El archivo `/etc/crontab` y todos los que se encuentran del directorio `/etc/cron.d` son crontabs del sistema.

La mayoría de las distribuciones también incluyen los directorios `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` y `/etc/cron.monthly` que contienen scripts para ser ejecutados con la frecuencia apropiada. Por ejemplo, si quiere ejecutar un script diariamente puede colocarlo en `/etc/cron.daily`.

WARNING	Algunas distribuciones usan `/etc/cron.d/hourly`, `/etc/cron.d/daily`, `/etc/cron.d/weekly` y `/etc/cron.d/monthly`. Recuerde siempre comprobar los directorios correctos donde colocar los scripts que quiere que cron ejecute.

La sintaxis de los crontabs del sistema es similar a la de los crontabs de los usuarios, sin embargo, también requiere un campo obligatorio adicional que especifica qué usuario ejecutará el trabajo de cron. Por lo tanto, cada línea de un crontab de sistema contiene siete campos separados por un espacio:

- El minuto de la hora (0-59).

- La hora del día (0-23).
- El día del mes (1-31).
- El mes del año (1-12).
- El día de la semana (0-7 con domingo=0 o domingo=7).
- El nombre de la cuenta de usuario que se utilizará al ejecutar el comando.
- El comando a ejecutar.

En cuanto a los crontabs de usuario, puede especificar múltiples valores para los campos de tiempo usando los operadores `*`, `,` , `-` y `/`. También puede indicar el mes y el día de la semana con las tres primeras letras del nombre en lugar del número correspondiente.

Especificaciones de tiempo particulares

Al editar los archivos crontab, también puede usar atajos especiales en las primeras cinco columnas en lugar de las especificaciones de tiempo:

`@reboot`
: Ejecutar la tarea especificada una vez después de reiniciar.

`@hourly`
: Ejecutar la tarea especificada una vez por hora al iniciar.

`@daily` (o **`@midnight`**)
: Ejecutar la tarea especificada una vez al día a medianoche.

`@weekly`
: Ejecutar la tarea especificada una vez a la semana a medianoche del domingo.

`@monthly`
: Ejecutar la tarea especificada una vez al mes a la medianoche del primer día del mes.

`@yearly` (o **`@annually`**)
: Ejecutar la tarea especificada una vez al año a medianoche del 1 de enero.

Variables de Crontab

En ocasiones, dentro de un archivo crontab, hay variables definidas antes de que se declaren las tareas programadas. Las variables de entorno establecidas (comúnmente) son:

`HOME`
: El directorio donde `cron` invoca los comandos (por defecto el directorio principal del usuario).

`MAILTO`
: El nombre del usuario o la dirección a la que se envía la salida estándar y el error (por defecto, el propietario del crontab). También se permiten múltiples valores separados por comas y un valor vacío indica que no se debe enviar ningún correo.

`PATH`
: La ubicación de los comandos en los sistemas de archivos.

`SHELL`
: El shell a usar (por defecto `/bin/sh`).

Crear trabajos en un cron de usuario

El comando `crontab` se usa para mantener los archivos `crontab` para usuarios individuales. En particular, puede ejecutar el comando `crontab -e` para editar su propio archivo crontab o para crear uno si aún no existe.

```
$ crontab -e
no crontab for frank - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        < ---- easiest
  3. /usr/bin/emacs24
  4. /usr/bin/vim.tiny

Choose 1-4 [2]:
```

Por defecto, el comando `crontab` abre el editor especificado por las variables de entorno `VISUAL` o `EDITOR` para que pueda empezar a editar su archivo `crontab` con su preferido. Algunas distribuciones, como se muestra en el ejemplo anterior, le permiten elegir el editor de una lista cuando `crontab` se ejecuta por primera vez.

Si quiere ejecutar el script `foo.sh` ubicado en su directorio principal todos los días a las 10:00 am, puede agregar la siguiente línea a su archivo crontab:

```
0 10 * * * /home/frank/foo.sh
```

Considere los siguientes ejemplos de entradas en el crontab:

```
0,15,30,45 08 * * 2 /home/frank/bar.sh
30 20 1-15 1,6 1-5 /home/frank/foobar.sh
```

En la primera línea, el script `bar.sh` se ejecuta todos los martes a las 08:00 am, a las 08:15 am, a las 08:30 am y a las 08:45 am. En la segunda, línea el script `foobar.sh` se ejecuta a las 08:30 pm de lunes a viernes durante los primeros quince días de enero y junio.

WARNING Aunque los archivos crontab pueden ser editados manualmente, siempre se recomienda usar el comando `crontab`. Los permisos de los archivos crontab normalmente sólo permiten su edición mediante el comando `crontab`.

Además de la opción `-e` mencionada anteriormente, el comando `crontab` tiene otras opciones útiles:

`-l`
: Muestra el crontab actual en la salida estándar.

`-r`
: Quita el crontab actual.

`-u`
: Especifica el nombre del usuario cuyo crontab necesita ser modificado. Esta opción requiere privilegios de root y permite que el usuario root edite los archivos crontab de otro usuario.

Crear crones de sistema

A diferencia de los crontabs de usuario, los crontabs de sistema se actualizan usando un editor: por lo tanto, no es necesario ejecutar el comando `crontab` para editar `/etc/crontab` y los archivos en `/etc/cron.d`. Recuerde que cuando edite los crontabs del sistema, debe especificar la cuenta que se usará para ejecutar el trabajo cron (normalmente el usuario root).

Por ejemplo, si quiere ejecutar el script `barfoo.sh` ubicado en el directorio `/root` todos los días a la 01:30 am, puede abrir `/etc/crontab` con su editor preferido y agregar la siguiente línea:

```
30 01 * * * root /root/barfoo.sh >>/root/output.log 2>>/root/error.log
```

En el ejemplo anterior, la salida del job se añade a `/root/output.log`, mientras que los errores se añaden a `/root/error.log`.

WARNING

A menos que la salida sea redirigida a un archivo como en el ejemplo anterior (o que la variable `MAILTO` se establezca en un valor vacío), toda la salida de un trabajo cron será enviada al usuario por correo electrónico. Una práctica común es redirigir la salida estándar a `/dev/null` (o a un archivo para su posterior revisión si es necesario) y no redirigir el error estándar. De esta manera el usuario será notificado inmediatamente por correo electrónico de cualquier error.

Configurar el acceso a la programación de tareas

En Linux los archivos `/etc/cron.allow` y `/etc/cron.deny` se usan para establecer las restricciones `crontab`. En particular, se usan para permitir o no la programación de trabajos cron para diferentes usuarios. Si existe el archivo `/etc/cron.allow`, sólo los usuarios no root listados dentro de él pueden programar trabajos cron usando el comando `crontab`. Si `/etc/cron.allow` no existe pero `/etc/cron.deny` existe, sólo los usuarios no root listados dentro de este archivo no pueden programar trabajos cron usando el comando `crontab` (en este caso un `/etc/cron.deny` vacío significa que a cada usuario se le permite programar trabajos cron con `crontab`). Si no existe ninguno de estos archivos, el acceso del usuario a la programación de trabajos cron dependerá de la distribución utilizada.

NOTE

Los archivos `/etc/cron.allow` y `/etc/cron.deny` contienen una lista de nombres de usuario, cada uno en una línea separada.

Una alternativa a cron

Usando systemd como el administrador del sistema y del servicio, puede establecer *timers* como una alternativa a `cron` para programar sus tareas. Los temporizadores son archivos de unidad systemd identificados por el sufijo `.timer`, y para cada uno de ellos debe haber un archivo de unidad correspondiente que describa la unidad que se activará cuando el temporizador transcurra. Por defecto, un `timer` activa un servicio con el mismo nombre, excepto por el sufijo.

Un temporizador incluye una sección de `[Timer]` que especifica cuándo deben ejecutarse los trabajos programados. Específicamente, puede usar la opción `OnCalendar=` para definir *temporizadores en tiempo real* que funcionan de la misma manera que los trabajos cron (están basados en expresiones de eventos de calendario). La opción `OnCalendar=` requiere la siguiente sintaxis:

```
DayOfWeek Year-Month-Day Hour:Minute:Second
```

Con el `DayOfWeek` siendo opcional. Los operadores `*`, `/` y `,` tienen el mismo significado que los usados para los trabajos de cron, mientras que puede usar `..` entre dos valores para indicar un rango contiguo. Para la especificación `DayOfWeek`, puede usar las tres primeras letras del nombre o el nombre completo.

> **NOTE** También se pueden definir *temporizadores monótonos* que se activan después de transcurrir un tiempo desde un punto de inicio específico (por ejemplo, cuando inicia máquina o cuando se activa el propio temporizador).

Por ejemplo, si quiere ejecutar el servicio llamado `/etc/systemd/system/foobar.service` a las 05:30 del primer lunes de cada mes, puede añadir las siguientes líneas en el archivo correspondiente de la unidad `/etc/systemd/system/foobar.timer`.

```
[Unit]
Description=Run the foobar service

[Timer]
OnCalendar=Mon *-*-1..7 05:30:00
Persistent=true

[Install]
WantedBy=timers.target
```

Una vez que haya creado el nuevo temporizador, puede activarlo e iniciarlo ejecutando los siguientes comandos como root:

```
# systemctl enable foobar.timer
# systemctl start foobar.timer
```

Puede cambiar la frecuencia de su trabajo programado, modificando el valor `OnCalendar` y luego escribiendo el comando `Systemctl daemon-reload`.

Finalmente, si quiere ver la lista de temporizadores activos ordenados por el tiempo que transcurre a continuación, puede usar el comando `systemctl list-timers`. Puede añadir la opción `--all` para ver también las unidades de temporizadores inactivos.

> **NOTE**
> Recuerda que los temporizadores se registran en el diario del sistema (system journal) y puede revisar los registros de las diferentes unidades usando el comando "journalalctl". Recuerde también que si esta utilizando un usuario ordinario, necesita usar la opción "user" de los comandos "systemctl" y "journalctl".

En lugar de la formas mencionadas anteriormente, se pueden utilizar algunas expresiones especiales que describen frecuencias particulares para la ejecución del trabajo:

`hourly`
Ejecutar la tarea especificada una vez por hora al comienzo de la hora.

`daily`
Ejecutar la tarea especificada una vez al día a medianoche.

`weekly`
Ejecutar la tarea especificada una vez a la semana a medianoche del lunes.

`monthly`
Ejecutar la tarea especificada una vez al mes a la medianoche del primer día del mes.

`yearly`
Ejecutar la tarea especificada una vez al año a medianoche del primer día de enero.

Puede ver las páginas de man para la lista completa de especificaciones de tiempo y fecha en `systemd.timer(5)`.

Ejercicios guiados

1. Para cada uno de los siguientes atajos `crontab`, indique la especificación de tiempo correspondiente (es decir, las cinco primeras columnas de un archivo `crontab` de usuario):

`@hourly`	
`@daily`	
`@weekly`	
`@monthly`	
`@annually`	

2. Para cada uno de los siguientes atajos `OnCalendar`, indique la especificación de tiempo correspondiente (la forma más larga):

`hourly`	
`daily`	
`weekly`	
`monthly`	
`yearly`	

3. Explique el significado de las siguientes especificaciones de tiempo que se encuentran en un archivo `crontab`:

`30 13 * * 1-5`	
`00 09-18 * * *`	
`30 08 1 1 *`	
`0,20,40 11 * * Sun`	
`00 09 10-20 1-3 *`	
`*/20 * * * *`	

4. Explique el significado de las siguientes especificaciones de tiempo utilizadas en la opción `OnCalendar` de un archivo de temporizador:

`*-*-* 08:30:00`	
`Sat,Sun *-*-* 05:00:00`	
`*-*-01 13:15,30,45:00`	
`Fri *-09..12-* 16:20:00`	

`Mon,Tue *-*-1,15 08:30:00`	
`*-*-* *:00/05:00`	

Ejercicios de exploración

1. Asumiendo que usted está autorizado a programar trabajos con `cron` como un usuario ordinario, ¿Qué comando usaría para crear su propio archivo crontab?

2. Cree un trabajo simple y programado que ejecute el comando `date` todos los viernes a la 01:00 pm. ¿Dónde puede ver la salida de este trabajo?

3. Cree otro trabajo programado que ejecute el script `foobar.sh` cada minuto, redirigiendo la salida al archivo `output.log` en su directorio de origen para que sólo se le envíe el error estándar por correo electrónico.

4. Mire la entrada `crontab` del nuevo trabajo programado. ¿Por qué no es necesario especificar la ruta absoluta del archivo en el que se guarda la salida estándar? ¿Y por qué puede usar el comando `./foobar.sh` para ejecutar el script?

5. Edite la entrada anterior `crontab` eliminando la redirección de salida y desactive el primer trabajo cron que había creado.

6. ¿Cómo puede enviar la salida y los errores de un trabajo programado a la cuenta de usuario `emma` por correo electrónico? ¿Y cómo puede evitar enviar la salida y los errores estándar por correo electrónico?

7. Ejecute el comando `ls -l /usr/bin/crontab`. ¿Qué bit especial se establece y cuál es su significado?

Resumen

En esta lección, aprendió:

- Usar `cron` para ejecutar trabajos a intervalos regulares.
- Administrar los trabajos de "cron".
- Configurar el acceso del usuario a la programación de trabajos cron.
- Comprender el papel de las unidades de tiempo del sistema como una alternativa a `cron`.

Los siguientes comandos y archivos fueron discutidos en esta lección:

`crontab`
: Mantener los archivos `crontab` para los usuarios individuales.

`/etc/cron.allow` y **`/etc/cron.deny`**
: Archivos particulares usados para establecer restricciones `crontab`.

`/etc/crontab`
: Archivo crontab del sistema.

`/etc/cron.d`
: El directorio que contiene los archivos crontab del sistema.

`systemctl`
: Controla el sistema y el administrador del servicio. En relación con los temporizadores, se puede utilizar para habilitarlos e iniciarlos.

Respuestas a los ejercicios guiados

1. Para cada uno de los siguientes atajos `crontab`, indique la especificación de tiempo correspondiente (es decir, las cinco primeras columnas de un archivo `crontab` de usuario):

`@hourly`	`0 * * * *`
`@daily`	`0 0 * * *`
`@weekly`	`0 0 * * 0`
`@monthly`	`0 0 1 * *`
`@annually`	`0 0 1 1 *`

2. Para cada uno de los siguientes atajos `OnCalendar`, indique la especificación de tiempo correspondiente (la forma más larga):

`hourly`	`*-*-* *:00:00`
`daily`	`*-*-* 00:00:00`
`weekly`	`Mon *-*-* 00:00:00`
`monthly`	`*-*-01 00:00:00`
`yearly`	`*-01-01 00:00:00`

3. Explique el significado de las siguientes especificaciones de tiempo para un archivo `crontab`:

`30 13 * * 1-5`	A las 13:30 horas todos los días de la semana de lunes a viernes
`00 09-18 * * *`	Todos los días y cada hora desde las 09:00 hasta las 18:00
`30 08 1 1 *`	A las 08:30 horas del primer día de enero
`0,20,40 11 * * Sun`	Todos los domingos a las 11:00, 11:20 y 11:40
`00 09 10-20 1-3 *`	A las 09:00 horas del 10 al 20 de enero, febrero y marzo
`*/20 * * * *`	Cada veinte minutos

4. Explique el significado de las siguientes especificaciones de tiempo para un archivo `crontab`:

`*-*-* 08:30:00`	Todos los días a las 08:30 am
`Sat,Sun *-*-* 05:00:00`	A las 05:00 am del sábado y el domingo

`*-*-01 13:15,30,45:00`	A las 01:15 pm, 01:30 pm y 01:45 pm del primer día del mes
`Fri *-09..12-* 16:20:00`	A las 04:20 pm todos los viernes de septiembre, octubre, noviembre y diciembre
`Mon,Tue *-*-1,15 08:30:00`	A las 8:30 de la mañana del primero o del decimoquinto día de cada mes, sólo si el día es un lunes o un martes.
`*-*-* *:00/05:00`	Cada cinco minutos

Respuestas a los ejercicios de exploración

1. Asumiendo que usted está autorizado a programar trabajos con `cron` como un usuario ordinario, ¿Qué comando usaría para crear su propio archivo crontab?

```
dave@hostname ~ $ crontab -e
no crontab for dave - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        < ---- easiest
  3. /usr/bin/emacs24
  4. /usr/bin/vim.tiny

Choose 1-4 [2]:
```

2. Cree un trabajo simple y programado que ejecute el comando `date` todos los viernes a la 01:00 pm. ¿Dónde puede ver la salida de este trabajo?

```
00 13 * * 5 date
```

La salida se envía por correo al usuario; para poder visualizarla, se puede utilizar el comando `mail`.

3. Cree otro trabajo programado que ejecute el script `foobar.sh` cada minuto, redirigiendo la salida al archivo `output.log` en su directorio de origen para que sólo se le envíe el error estándar por correo electrónico.

```
*/1 * * * * ./foobar.sh >> output.log
```

4. Mire la entrada `crontab` del nuevo trabajo programado. ¿Por qué no es necesario especificar la ruta absoluta del archivo en el que se guarda la salida estándar? ¿Y por qué puede usar el comando `./foobar.sh` para ejecutar el script?

`cron` invoca los comandos desde el directorio home del usuario, a menos que se especifique otra ubicación por la variable de entorno `HOME` dentro del archivo `crontab`. Por esta razón, puede utilizar la ruta relativa del archivo de salida y ejecutar el script con `./foobar.sh`.

5. Edite la entrada anterior `crontab` eliminando la redirección de salida y desactive el primer trabajo cron que había creado.

```
#00 13 * * 5 date
*/1 * * * * ./foobar.sh
```

Para deshabilitar un trabajo cron, puede simplemente comentar la línea correspondiente dentro del archivo `crontab`.

6. ¿Cómo puede enviar la salida y los errores de un trabajo programado a la cuenta de usuario `emma` por correo electrónico? ¿Y cómo puede evitar enviar la salida y los errores estándar por correo electrónico?

   Para enviar la salida estándar y el error a `emma`, debe establecer la variable de entorno `MAILTO` en su archivo `crontab` de la siguiente manera:

   ```
   MAILTO="emma"
   ```

   Para especificarle a `cron` que no se debe enviar ningún correo, puede asignar un valor vacío a la variable de entorno `MAILTO`.

   ```
   MAILTO=""
   ```

7. Ejecute el comando `ls -l /usr/bin/crontab`. ¿Qué bit especial se establece y cuál es su significado?

   ```
   $ ls -l /usr/bin/crontab
   -rwxr-sr-x 1 root crontab 25104 feb 10  2015 /usr/bin/crontab
   ```

   El comando `crontab` tiene el bit SGID establecido (el caracter `s` en lugar del flag ejecutable para el grupo), lo que significa que se ejecuta con los privilegios del grupo (por lo tanto `crontab`). Es por esto que los usuarios comunes pueden editar su archivo `crontab` usando el comando `crontab`. Tenga en cuenta que muchas distribuciones tienen permisos de archivo establecidos de tal manera que los archivos `crontab` sólo pueden ser editados mediante el comando `crontab`.



107.2 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	107 Tareas administrativas
Objetivo:	107.2 Automatizar tareas administrativas del sistema mediante la programación de trabajos
Lección:	2 de 2

Introducción

Como aprendimos en la lección anterior, puede programar tareas regulares usando crones, pero a veces puede necesitar ejecutar una tarea en un momento específico en el futuro. Para ello, es posible usar otra poderosa utilidad: el comando `at`.

Programar tareas con `at`

El comando `at` se utiliza para la programación de tareas una única vez y sólo requiere que se especifique cuándo se deba ejecutar una tarea en el futuro. Después de introducir `at` en la línea de comandos, seguido de la especificación de tiempo, entrará en la línea de comandos `at` donde puede definir los comandos a ejecutar. Puede salir del prompt con la secuencia de teclas Ctrl + D.

```
$ at now +5 minutes
warning: commands will be executed using /bin/sh
at> date
at> Ctrl+D
job 12 at Sat Sep 14 09:15:00 2019
```

El ejemplo anterior simplemente ejecuta el comando `date` después de cinco minutos. Similar a `cron`, la salida estándar y el error se envía por correo electrónico. Tenga en cuenta que el demonio `atd` tendrá que estar ejecutándose en el sistema para que pueda usar la programación de `at`.

> **NOTE** En Linux, el comando `batch` es similar a `at`, sin embargo las tareas `batch` se ejecutan sólo cuando la carga del sistema es lo suficientemente baja como para permitirlo.

Las opciones más importantes que se aplican al comando `at` son:

`-c`
: Imprime los comandos de una tarea específica (por medio del ID) a la salida estándar.

`-d`
: Borra las tareas basadas en su ID. Es un alias para `atrm`.

`-f`
: Lee las tareas desde un archivo en lugar de la entrada estándar.

`-l`
: Lista las tareas pendientes del usuario. Si el usuario es root, se listan todas las tareas de todos los usuarios. Es un alias para `atq`.

`-m`
: Envía un correo al usuario al final de la tarea aunque no haya mostrado salida.

`-q`
: Especifica una cola en forma de una sola letra de `a` a `z` y de `A` a `Z` (por defecto `a` para `at` y `b` para `batch`). Las tareas en las colas con las letras más altas se ejecutan con mayor prioridad. Los trabajos enviados a una cola con mayúsculas son tratados como tareas `batch`.

`-v`
: Muestra el tiempo en el que la tarea se ejecutará antes de leerla.

Listar tareas programadas con `atq`

Ahora programemos dos tareas más: la primera ejecuta el script `foo.sh` a las 09:30 am, mientras que la segunda ejecuta el script `bar.sh` después de una hora.

```
$ at 09:30 AM
warning: commands will be executed using /bin/sh
at> ./foo.sh
at> Ctrl+D
job 13 at Sat Sep 14 09:30:00 2019
$ at now +2 hours
warning: commands will be executed using /bin/sh
at> ./bar.sh
at> Ctrl+D
job 14 at Sat Sep 14 11:10:00 2019
```

Para listar sus tareas pendientes, puede usar el comando `atq` que muestra la siguiente información para cada tarea: ID, fecha de ejecución, tiempo de ejecución, cola y nombre de usuario.

```
$ atq
14      Sat Sep  14 11:10:00 2019 a frank
13      Sat Sep  14 09:30:00 2019 a frank
12      Sat Sep  14 09:15:00 2019 a frank
```

Recuerde que el comando `at -l` es un alias para `atq`.

NOTE Si ejecuta `atq` como root, mostrará las tareas en cola para todos los usuarios.

Borrar tareas con `atrm`

Si desea borrar una tarea de `at`, puede usar el comando `atrm` seguido del ID de la tarea. Por ejemplo, para borrar el trabajo con ID 14, puede ejecutar lo siguiente:

```
$ atrm 14
```

Puede borrar múltiples trabajos con `atrm` especificando múltiples identificaciones separadas por espacios. Recuerde que el comando `at -d` es un alias para `atrm`.

NOTE Si ejecuta `atrm` como root puede borrar los trabajos de todos los usuarios.

Configurar el acceso a la programación de tareas

La autorización para que los usuarios ordinarios programen las tareas en `at` está determinada por los archivos `/etc/at.allow` y `/etc/at.deny`. Si existe el archivo `etc/at.allow`, sólo los usuarios que no

son root listados en este pueden programar tareas `at`. Si no existe `/etc/at.allow` pero sí `/etc/at.deny`, los usuarios que no son root listados en este no podrán programar tareas en `at` (un archivo vacío `/etc/at.deny` significa que cada usuario puede programar trabajos `at`). Si ninguno de estos archivos existe, el acceso del usuario a la programación de trabajos `at` depende de la distribución utilizada.

Especificaciones de tiempo

Puede especificar cuándo ejecutar una determinada tarea con `at` utilizando el formato `HH:MM`, opcionalmente seguido de AM o PM en caso de formato de 12 horas. Si la hora especificada ya ha pasado, se asume el día siguiente. Si quiere programar una fecha particular en la que se ejecutará el trabajo, debe añadir la información de la fecha después de la hora usando uno de los siguientes formatos: `mes día-de-mes`, `mes día-de-mes año`, `MMDDYY`, `MM/DD/YY`, `DD.MM.YY` y `YYYY-MM-DD`).

También se aceptan las siguientes palabras claves: `midnight`, `noon`, `teatime` (4 pm) y `now` seguido de un signo más (+) y un período de tiempo (minutos, horas, días y semanas). Por último, puede indicarle a `at` que ejecute el trabajo hoy o mañana sufijando la hora con las palabras `today` o `tomorrow`. Por ejemplo, puede usar `at 07:15 AM Jan 01` para ejecutar un trabajo a las 07:15 AM del 1 de enero y `at now +5 minutes` para ejecutar un trabajo dentro de 5 minutos. Puede leer el archivo `Timespec` en el directorio `/usr/share` para más información sobre la definición exacta de las especificaciones de tiempo.

Una alternativa a `at`

Usando systemd como el administrador del sistema y de servicio, también puede programar tareas únicas con el comando `systemd-run`. Normalmente se utiliza para crear una unidad de temporizador transitoria de modo que un comando se ejecute en un momento específico sin necesidad de crear un archivo de servicio. Por ejemplo, actuando como root, puede ejecutar el comando `date` a las 11:30 AM el 2019/10/06 usando el siguiente comando:

```
# systemd-run --on-calendar='2019-10-06 11:30' date
```

Si quiere ejecutar el script `foo.sh`, ubicado en su directorio de trabajo actual, después de dos minutos puede usar:

```
# systemd-run --on-active="2m" ./foo.sh
```

Consulte las páginas de man para aprender todos los usos posibles de `systemd-run` con `systemd-run(1)`.

> **NOTE**
>
> Recuerde que los temporizadores se registran en el diario del sistema y puede revisar los registros de las diferentes unidades usando el comando `journalalctl`. Recuerde también que si está actuando como un usuario ordinario, necesita usar la opción `--user` de los comandos `systemd-run` y `journalctl`.

Ejercicios guiados

1. Para cada una de las siguientes especificaciones de tiempo, indique cuál es válida y cuál no lo es para `at`:

`at 08:30 AM next week`	
`at midday`	
`at 01-01-2020 07:30 PM`	
`at 21:50 01.01.20`	
`at now +4 days`	
`at 10:15 PM 31/03/2021`	
`at tomorrow 08:30 AM`	

2. Una vez que ha programado una tarea con `at`, ¿cómo puede revisar sus comandos?

3. ¿Qué comandos puede usar para revisar tu trabajos pendientes? ¿Qué comandos usaría para borrarlos?

4. Con systemd, ¿qué comando se utiliza como alternativa a `at`?

Ejercicios de exploración

1. Cree un trabajo que ejecute el script `foo.sh` ubicado en su directorio personal, a las 10:30 am del próximo 31 de octubre. Asuma que está actuando como un usuario ordinario.

2. Entre en el sistema como otro usuario ordinario y cree otra tarea con `at` que ejecute el script `bar.sh` mañana a las 10:00 am. Supongamos que el script se encuentra en el directorio principal del usuario.

3. Entra en el sistema como otro usuario ordinario y crea otra tarea con `at` que ejecute el script `foobar.sh` justo después de 30 minutos. Supongamos que el script se encuentra en el directorio principal del usuario.

4. Ahora como root, ejecute el comando `atq` para revisar los trabajos `at` programados de todos los usuarios. ¿Qué pasa si un usuario ordinario ejecuta este comando?

5. Como root, borre todos estos trabajos pendientes en `at` usando un solo comando.

6. Ejecute el comando `ls -l /usr/bin/at` y examine sus permisos.

Resumen

En esta lección, aprendió:

- Usar `at` para ejecutar trabajos una única vez en un momento específico.
- Administrar las tareas en `at`.
- Configurar el acceso de los usuarios a la programación de las tareas.
- Utilizar `systemd-run` como alternativa a `at`.

Los siguientes comandos y archivos fueron discutidos en esta lección:

`at`
: Ejecuta comandos en un momento determinado.

`atq`
: Muestra las tareas pendientes del usuario, a menos que el usuario sea el superusuario.

`atrm`
: Borra las tareas de `at`, identificadas por su número de trabajo.

`/etc/at.allow` y `/etc/at.deny`
: Archivos particulares usados para establecer restricciones de `at`.

`systemd-run`
: Crea e inicia una unidad transitoria de `tiempo` como alternativa de `at` para la programación de una única vez.

Respuestas a los ejercicios guiados

1. Para cada una de las siguientes especificaciones de tiempo, indique cuál es válido y cuál no lo es para `at`:

`at 08:30 AM next week`	Válido
`at midday`	Inválido
`at 01-01-2020 07:30 PM`	Inválido
`at 21:50 01.01.20`	Válido
`at now +4 days`	Válido
`at 10:15 PM 31/03/2021`	Inválido
`at tomorrow 08:30 AM monotonic`	Inválido

2. Una vez que ha programado una tarea con `at`, ¿cómo puede revisar sus comandos?

   Puede usar el comando `at -c` seguido del ID de la tarea cuyos comandos quiere revisar. Tenga en cuenta que la salida también contiene la mayor parte del entorno que estaba activo en el momento en que se programó el trabajo. Recuerde que root puede revisar los trabajos de todos los usuarios.

3. ¿Qué comandos puede usar para revisar sus trabajos pendientes? ¿Qué comandos usaría para borrarlos?

   Puede usar el comando `at -l` para revisar sus trabajos pendientes, y puede usar el comando `at -d` para borrarlos. `at -l` es un alias para `atq` y `at -d` es un alias para `atrm`. Recuerde que root puede listar y borrar las tareas de todos los usuarios.

4. Con systemd, ¿qué comando se utiliza como alternativa a `at`?

   El comando `systemd-run` puede ser usado como una alternativa a `at` para programar trabajos de una sola vez. Por ejemplo, se puede usar para ejecutar comandos a una hora específica, definiendo un *temporizador de calendario* o un *temporizador monótono* relativo a diferentes puntos de inicio.

Respuestas a los ejercicios de exploración

1. Cree un trabajo que ejecute el script `foo.sh` ubicado en su directorio personal, a las 10:30 am del próximo 31 de octubre. Asuma que está actuando como un usuario ordinario.

```
$ at 10:30 AM October 31
warning: commands will be executed using /bin/sh
at> ./foo.sh
at> Ctrl+D
job 50 at Thu Oct 31 10:30:00 2019
```

2. Entre en el sistema como otro usuario ordinario y cree otra tarea con `at` que ejecute el script `bar.sh` mañana a las 10:00 am. Supongamos que el script se encuentra en el directorio principal del usuario.

```
$ at 10:00 AM tomorrow
warning: commands will be executed using /bin/sh
at> ./bar.sh
at> Ctrl+D
job 51 at Sun Oct 6 10:00:00 2019
```

3. Entre en el sistema como otro usuario ordinario y cree otra tarea con `at` que ejecute el script `bar.sh` mañana a las 10:00 am. Supongamos que el script se encuentra en el directorio principal del usuario.

```
$ at now +30 minutes
warning: commands will be executed using /bin/sh
at> ./foobar.sh
at> Ctrl+D
job 52 at Sat Oct 5 10:19:00 2019
```

4. Ahora como root, ejecute el comando `atq` para revisar los trabajos `at` programados de todos los usuarios. ¿Qué pasa si un usuario ordinario ejecuta este comando?

```
# atq
52      Sat Oct  5 10:19:00 2019 a dave
50      Thu Oct 31 10:30:00 2019 a frank
51      Sun Oct  6 10:00:00 2019 a emma
```

Si ejecuta el comando `atq` como root, todos las tareas de `at` pendientes de todos los usuarios aparecen en la lista. Si lo ejecuta como un usuario ordinario, sólo sus propios trabajos pendientes `at` serán listados.

5. Como root, borre todos estos trabajos pendientes en `at` usando un solo comando.

```
# atrm 50 51 52
```

6. Ejecute el comando `ls -l /usr/bin/crontab`. ¿Qué bit especial se establece y cuál es su significado?

```
# ls -l /usr/bin/at
-rwsr-sr-x 1 daemon daemon 43762 Dec  1  2015 /usr/bin/at
```

En esta distribución, el comando `at` tiene establecidos los bits SUID (el caracter `s` en lugar de la marca ejecutable para el propietario) y SGID (el caracter `s` en lugar de la marca ejecutable para el grupo), lo que significa que se ejecuta con los privilegios del propietario y del grupo del archivo (`daemon` para ambos). Es por eso que los usuarios comunes pueden programar trabajos con `at`.



107.3 Localización e internacionalización

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 107.3

Importancia

3

Áreas de conocimiento clave

- Configuración regional y variables de entorno.
- Configuración de la zona horaria y de las variables de entorno.

Lista parcial de archivos, términos y utilidades

- `/etc/timezone`
- `/etc/localtime`
- `/usr/share/zoneinfo/`
- `LC_*`
- `LC_ALL`
- `LANG`
- `TZ`
- `/usr/bin/locale`
- `tzselect`
- `timedatectl`
- `date`

- `iconv`
- `UTF-8`
- `ISO-8859`
- ASCII
- Unicode



107.3 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	107 Tareas administrativas
Objetivo:	107.3 Localización e internacionalización
Lección:	1 de 1

Introducción

Todas las principales distribuciones de Linux pueden ser configuradas para utilizar ajustes de localización personalizados. Estos ajustes incluyen definiciones relacionadas con la región y el idioma, como la zona horaria, el idioma de la interfaz, así como la codificación de caracteres que pueden ser modificados durante la instalación del sistema operativo o en cualquier momento posterior.

Las aplicaciones se basan en variables de entorno, archivos de configuración del sistema y comandos para seleccionar la hora y el idioma adecuados; por lo tanto, la mayoría de las distribuciones comparten una forma estandarizada de ajustar la hora y los ajustes de localización. Estos ajustes son importantes no sólo para mejorar la experiencia del usuario, sino también para asegurar que la hora de los eventos importantes del sistema se calcule correctamente, por ejemplo, informar sobre temas relacionados con la seguridad .

Para poder representar cualquier texto escrito, independientemente del idioma hablado, los sistemas operativos modernos necesitan una referencia *estándar de codificación de caracteres*, y los sistemas Linux no son la excepción. Como las computadoras sólo pueden tratar con números, un caracter de texto no es más que un número asociado a un símbolo gráfico. Distintas plataformas informáticas

pueden asociar valores numéricos distintos al mismo caracter, por lo que se necesita una norma de codificación de caracteres común para que sean compatibles. Un documento de texto creado en un sistema sólo será legible en otro sistema si ambos coinciden en el formato de codificación y en qué número se asocia a qué carácter, o al menos si saben cómo convertirlo entre las dos normas.

La naturaleza heterogénea de los ajustes de localización en los sistemas basados en Linux da lugar a sutiles diferencias entre las distribuciones. A pesar de estas diferencias, todas las distribuciones comparten las mismas herramientas y conceptos básicos para configurar los aspectos de internacionalización de un sistema.

Zonas horarias

Las zonas horarias son bandas discretas de la superficie de la Tierra que abarcan el equivalente a una hora, es decir, regiones del mundo que experimentan la misma hora del día en un momento dado. Como no hay una sola longitud que pueda considerarse como el comienzo del día para todo el mundo, las zonas horarias son relativas al *meridiano principal*, donde el ángulo de longitud de la Tierra se define como 0. La hora en el meridiano principal se denomina *Hora Universal Coordinada*, por convención abreviada como UTC. Por razones prácticas, los husos horarios no siguen la distancia longitudinal exacta del punto de referencia (el meridiano principal). En su lugar, los husos horarios se adaptan artificialmente para seguir las fronteras de los países u otras subdivisiones importantes.

Las subdivisiones políticas son tan relevantes que las zonas horarias reciben el nombre de algún agente geográfico importante de esa zona en particular, normalmente basado en el nombre de un gran país o ciudad dentro de la zona. Sin embargo, los husos horarios se dividen según su desfase horario en relación con el UTC y este desfase también puede utilizarse para indicar la zona en cuestión. La zona horaria *GMT-5*, por ejemplo, indica una región cuya hora UTC está cinco horas adelantada, es decir, esa región está cinco horas atrasada respecto de la UTC. Asimismo, el huso horario *GMT+3* indica una región para la cual la hora UTC está tres horas por detrás. El término GMT - de *Greenwich Mean Time* - se utiliza como sinónimo de UTC en los nombres de las zonas basadas en la compensación.

Se puede acceder a una máquina conectada desde diferentes partes del mundo, por lo que es una buena práctica ajustar el reloj del hardware a UTC (la zona horaria GMT+0) y dejar la elección de la zona horaria a cada caso particular. Los servicios en la nube, por ejemplo, se configuran comúnmente para usar UTC, ya que puede ayudar a mitigar las inconsistencias ocasionales entre la hora local y la hora de los clientes o en otros servidores. Por el contrario, los usuarios que abren una sesión remota en el servidor pueden querer utilizar su zona horaria local. Por lo tanto, dependerá del sistema operativo establecer la zona horaria correcta según cada caso.

Además de la fecha y la hora actuales, el comando `date` también imprimirá la zona horaria actualmente configurada:


```
$ date
Mon Oct 21 10:45:21 -03 2019
```

El desplazamiento relativo al UTC viene dado por el valor `-03`, lo que significa que la hora mostrada tiene 3 horas de retraso con respecto a UTC. Por lo tanto, la hora UTC está tres horas por delante, haciendo que *GMT-3* sea el huso horario correspondiente a la hora dada. El comando `timedatectl`, que está disponible en las distribuciones que utilizan systemd, muestra más detalles sobre la hora y la fecha del sistema:

```
$ timedatectl
                      Local time: Sat 2019-10-19 17:53:18 -03
                  Universal time: Sat 2019-10-19 20:53:18 UTC
                        RTC time: Sat 2019-10-19 20:53:18
                       Time zone: America/Sao_Paulo (-03, -0300)
       System clock synchronized: yes
systemd-timesyncd.service active: yes
                 RTC in local TZ: no
```

Como se muestra en la entrada `Time zone`, los nombres de las zonas horarias basados en localidades - como `America/Sao_Paulo` - también se aceptan. La zona horaria por defecto del sistema se mantiene en el archivo `/etc/timezone`, ya sea por el nombre descriptivo completo de la zona o por la diferencia de horas. Los nombres genéricos de la zona horaria dados por la diferencia de horas UTC deben incluir `Etc` como la primera parte del nombre. Así que para fijar la zona horaria por defecto en GMT+3, el nombre de la zona horaria debe ser `Etc/GMT+3`:

```
$ cat /etc/timezone
Etc/GMT+3
```

Aunque los nombres de las zonas horarias basados en las localidades no requieren el desplazamiento de la hora para funcionar, no es tan sencillo elegirlos. La misma zona puede tener más de un nombre, lo que puede ser difícil de recordar. Para facilitar esto, el comando `tzselect` ofrece un método interactivo que guiará al usuario hacia la definición correcta de la zona horaria. El comando `tzselect` debería estar disponible por defecto en todas las distribuciones de Linux, ya que lo proporciona el paquete que contiene los programas de utilidades necesarios relacionados con la Biblioteca C de GNU.

El comando `tzselect` será útil, por ejemplo, para un usuario que quiera identificar la zona horaria de "São Paulo City" en "Brazil". El comando `tzselect` comienza preguntando por la macro región de la ubicación deseada:

```
$ tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
 1) Africa
 2) Americas
 3) Antarctica
 4) Asia
 5) Atlantic Ocean
 6) Australia
 7) Europe
 8) Indian Ocean
 9) Pacific Ocean
10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the time zone using the Posix TZ format.
#? 2
```

La opción `2` es para las regiones de América (Norte y Sur), no necesariamente en la misma zona horaria. También es posible especificar el huso horario con coordenadas geográficas o con la notación de desplazamiento, también conocido como el formato *Posix TZ format*. El siguiente paso es elegir el país:

```
Please select a country whose clocks agree with yours.
 1) Anguilla              19) Dominican Republic    37) Peru
 2) Antigua & Barbuda     20) Ecuador               38) Puerto Rico
 3) Argentina             21) El Salvador           39) St Barthelemy
 4) Aruba                 22) French Guiana         40) St Kitts & Nevis
 5) Bahamas               23) Greenland             41) St Lucia
 6) Barbados              24) Grenada               42) St Maarten (Dutch)
 7) Belize                25) Guadeloupe            43) St Martin (French)
 8) Bolivia               26) Guatemala             44) St Pierre & Miquelon
 9) Brazil                27) Guyana                45) St Vincent
10) Canada                28) Haiti                 46) Suriname
11) Caribbean NL          29) Honduras              47) Trinidad & Tobago
12) Cayman Islands        30) Jamaica               48) Turks & Caicos Is
13) Chile                 31) Martinique            49) United States
14) Colombia              32) Mexico                50) Uruguay
15) Costa Rica            33) Montserrat            51) Venezuela
16) Cuba                  34) Nicaragua             52) Virgin Islands (UK)
17) Curaçao               35) Panama                53) Virgin Islands (US)
18) Dominica              36) Paraguay
#? 9
```


El territorio de Brasil abarca cuatro zonas horarias, por lo que la información del país por sí sola no es suficiente para establecer la zona horaria. En el siguiente paso, el comando `tzselect` requerirá que el usuario especifique la región local:

```
Please select one of the following time zone regions.
 1) Atlantic islands
 2) Pará (east); Amapá
 3) Brazil (northeast: MA, PI, CE, RN, PB)
 4) Pernambuco
 5) Tocantins
 6) Alagoas, Sergipe
 7) Bahia
 8) Brazil (southeast: GO, DF, MG, ES, RJ, SP, PR, SC, RS)
 9) Mato Grosso do Sul
10) Mato Grosso
11) Pará (west)
12) Rondônia
13) Roraima
14) Amazonas (east)
15) Amazonas (west)
16) Acre
#? 8
```

No están disponibles todos los nombres de las localidades, pero elegir la región más cercana será suficiente. La información dada será utilizada por `tzselect` para mostrar la zona horaria correspondiente:

```
Se ha dado la siguiente información:

        Brazil
        Brazil (southeast: GO, DF, MG, ES, RJ, SP, PR, SC, RS)

Therefore TZ='America/Sao_Paulo' will be used.
Selected time is now:   sex out 18 18:47:07 -03 2019.
Universal Time is now:  sex out 18 21:47:07 UTC 2019.
Is the above information OK?
1) Yes
2) No
#? 1

You can make this change permanent for yourself by appending the line
        TZ='America/Sao_Paulo'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
America/Sao_Paulo
```

El nombre de la zona horaria resultante, `America/Sao_Paulo`, también puede ser usado como el contenido del fichero `/etc/timezone` para informar la zona horaria por defecto del sistema:

```
$ cat /etc/timezone
America/Sao_Paulo
```

Como se indica en la salida de `tzselect`, la variable de entorno `TZ` define la zona horaria de la sesión de shell, sea cual sea la zona horaria (por defecto) del sistema. Añadiendo la línea `TZ='America/Sao_Paulo'; export TZ` al archivo `~/.profile` hará que `America/Sao_Paulo` sea la zona horaria para las futuras sesiones del usuario. La variable `TZ` también puede ser modificada temporalmente durante la sesión actual, para mostrar la hora en una zona horaria diferente:

```
$ env TZ='Africa/Cairo' date
Mon Oct 21 15:45:21 EET 2019
```

En el ejemplo, `env` ejecutará el comando dado en una nueva sesión de sub-shell con las mismas variables de entorno de la sesión actual, excepto la variable `TZ`, modificado por el argumento `TZ='Africa/Cairo'`.

Horario de verano (Daylight Saving Time)

Muchas regiones adoptan un horario de ahorro de luz diurna (Daylight Saving Time) durante una parte del año — cuando los relojes se ajustan típicamente por una hora — que podría llevar a un sistema mal configurado a reportar la hora equivocada durante esa estación del año.

El archivo `/etc/localtime` contiene los datos utilizados por el sistema operativo para ajustar su hora en consecuencia. Los sistemas estándares de Linux tienen archivos para todas las zonas horarias en el directorio `/usr/share/zoneinfo/`, así que `/etc/localtime` es sólo un enlace simbólico al archivo de datos reales dentro de ese directorio. Los archivos en `/usr/share/zoneinfo/` están organizados por el nombre de la zona horaria correspondiente, así que el archivo de datos para la zona horaria `America/Sao_Paulo` será `/usr/share/zoneinfo/America/Sao_Paulo`

Como las definiciones para el horario de verano pueden cambiar, es importante mantener actualizados los archivos en `/usr/share/zoneinfo/`. El comando de actualización de la herramienta de gestión de paquetes que proporciona la distribución debería actualizarlos cada vez que haya una nueva versión disponible.

Lenguaje y codificación de caracteres

Los sistemas Linux pueden trabajar con una amplia variedad de lenguajes y codificaciones de caracteres no occidentales, definiciones conocidas como *locales*. La configuración de locale más básica es la definición de la variable de entorno `LANG`, a partir de la cual la mayoría de los programas de shell identifican el lenguaje a utilizar.

El contenido de la variable `LANG` sigue el formato `ab_CD`, donde `ab` es el código del idioma y `CD` es el código de la región. El código del idioma debe seguir la norma ISO-639 y el código de la región debe seguir la norma ISO-3166. Un sistema configurado para usar portugués brasileño, por ejemplo, debe tener la variable `LANG` definida como `pt_BR.UTF-8`:

```
$ echo $LANG
pt_BR.UTF-8
```

Como se ve en el ejemplo anterior, la variable `LANG` también contiene la codificación de caracteres prevista para el sistema. ASCII, abreviatura de *American Standard Code for Information Interchange*, fue la primera norma de codificación de caracteres ampliamente utilizada para la comunicación electrónica. Sin embargo, ASCII tiene un rango muy limitado de valores numéricos disponibles y dado que se basa en el alfabeto inglés, no contiene caracteres utilizados por otros idiomas o un conjunto ampliado de símbolos no alfabéticos. La codificación UTF-8 es una *Norma de Unicode* para los caracteres occidentales ordinarios, además de muchos otros símbolos no convencionales. Como ha señalado el *Consorcio del Unicode*, que mantiene el *Estándar del Unicode*, debe adoptarse por

defecto para garantizar la compatibilidad entre las plataformas informáticas:

> El Estándar Unicode proporciona un número único para cada caracter, sin importar la plataforma, el dispositivo, la aplicación o el idioma. Ha sido adoptado por todos los proveedores de software moderno y ahora permite que los datos sean transportados a través de muchas plataformas, dispositivos y aplicaciones diferentes sin corrupción. El soporte de Unicode constituye la base para la representación de idiomas y símbolos en todos los sistemas operativos principales, motores de búsqueda, navegadores, ordenadores portátiles y teléfonos inteligentes, además de Internet y la World Wide Web (URL, HTML, XML, CSS, JSON, etc.). (…) el estándar Unicode y la disponibilidad de herramientas que lo respaldan se encuentran entre las tendencias mundiales más importantes de la tecnología de software.
>
> — El Consorcio Unicode, ¿Qué es Unicode?

Algunos sistemas todavía pueden usar los estándares definidos por la ISO — como el estándar ISO-8859-1 — para la codificación de caracteres no ASCII. Sin embargo, tales estándares de codificación de caracteres deberían ser desaprobados en favor de los estándares de codificación de Unicode. No obstante, todos los principales sistemas operativos tienden a adoptar dicho estándar por defecto.

Los ajustes de localización del sistema están configurados en el archivo `/etc/locale.conf`. La variable `LANG` y otras variables relacionadas con la localización se asignan en este archivo como una variable de shell ordinaria, por ejemplo:

```
$ cat /etc/locale.conf
LANG=pt_BR.UTF-8
```

Los usuarios pueden usar una configuración de locale personalizada redefiniendo la variable de entorno `LANG`. Puede hacerse sólo para la sesión actual o para futuras sesiones, añadiendo la nueva definición al perfil del usuario en Bash, `~/.bash_profile` o `~/.profile`. Sin embargo, hasta que el usuario inicie la sesión, la localización del sistema por defecto seguirá siendo utilizada por programas independientes del usuario, como la pantalla de inicio de sesión del administrador de pantalla.

TIP El comando `localectl` disponible en los sistemas que emplean *systemd* como gestor del sistema, también puede ser usado para consultar y cambiar el locale del sistema. Por ejemplo: `localectl set-locale LANG=es_US.UTF-8`.

Además de la variable `LANG`, otras variables de entorno afectan aspectos específicos de la región, como el símbolo de la moneda a utilizar o el separador correcto de miles para los números:

`LC_COLLATE`
Establece el orden alfabético. Uno de sus propósitos es definir el orden en que los archivos y directorios son listados.

`LC_CTYPE`
Establece cómo el sistema tratará ciertos conjuntos de caracteres. Define, por ejemplo, qué caracteres considera como *mayúsculas* o *minúsculas*.

`LC_MESSAGES`
Establece el lenguaje a mostrar para los mensajes del programa (en su mayoría programas GNU).

`LC_MONETARY`
Establece la unidad monetaria y el formato de la moneda.

`LC_NUMERIC`
Establece el formato numérico para los valores no monetarios. Su principal propósito es definir los separadores de miles y decimales.

`LC_TIME`
Establece el formato de la hora y la fecha.

`LC_PAPER`
Establece el tamaño de papel estándar.

`LC_ALL`
Anula todas las demás variables, incluyendo `LANG`.

El comando `locale` mostrará todas las variables definidas en la configuración de locale actual:

```
$ locale
LANG=pt_BR.UTF-8
LC_CTYPE="pt_BR.UTF-8"
LC_NUMERIC=pt_BR.UTF-8
LC_TIME=pt_BR.UTF-8
LC_COLLATE="pt_BR.UTF-8"
LC_MONETARY=pt_BR.UTF-8
LC_MESSAGES="pt_BR.UTF-8"
LC_PAPER=pt_BR.UTF-8
LC_NAME=pt_BR.UTF-8
LC_ADDRESS=pt_BR.UTF-8
LC_TELEPHONE=pt_BR.UTF-8
LC_MEASUREMENT=pt_BR.UTF-8
LC_IDENTIFICATION=pt_BR.UTF-8
LC_ALL=
```

La única variable no definida es `LC_ALL`, que puede ser usada para anular temporalmente todos los ajustes de localización. El siguiente ejemplo muestra cómo el comando `date` — ejecutándose en un sistema configurado con locale `pt_BR.UTF-8` — modificará su salida para cumplir con la nueva variable `LC_ALL`:

```
$ date
seg out 21 10:45:21 -03 2019
$ env LC_ALL=en_US.UTF-8 date
Mon Oct 21 10:45:21 -03 2019
```

La modificación de la variable `LC_ALL` hizo que las abreviaturas del día de la semana y del mes se mostraran en inglés americano (`en_US`). Sin embargo, no es obligatorio establecer la misma localidad para todas las variables. Es posible, por ejemplo, hacer que el lenguaje definido a `pt_BR` y el formato numérico (`LC_NUMERIC`) se establezca en el estándar americano.

Algunos ajustes de localización cambian la forma en que los programas tratan el orden alfabético y los formatos de los números. Mientras que los programas convencionales suelen estar preparados para elegir correctamente una localización común para estas situaciones, los guiones pueden comportarse de forma inesperada cuando intentan ordenar correctamente por orden alfabético una lista de elementos. Por esta razón, se recomienda establecer la variable de entorno `LANG` en la localización común `C`, como en `LANG=C`, para que el script produzca resultados inequívocos, independientemente de las definiciones de localización utilizadas en el sistema donde se ejecute. El locale C sólo realiza una simple comparación bytewise, por lo que también tendrá un mejor rendimiento que los otros.

Conversión de la codificación

El texto puede aparecer con caracteres ininteligibles cuando se muestra con una codificación de caracteres distinta del sistema en el que se creó el texto. El comando `iconv` puede ser usado para resolver este problema, convirtiendo el archivo de su codificación de caracteres original a la deseada. Por ejemplo, para convertir un archivo llamado "original.txt" de la codificación ISO-8859-1 al archivo llamado "converted.txt" usando la codificación UTF-8, se puede usar el siguiente comando:

```
$ iconv -f ISO-8859-1 -t UTF-8 original.txt > converted.txt
```

La opción `-f ISO-8859-1` (o `--from-code=ISO-8859-1`) establece la codificación del archivo original y la opción `-t UTF-8` (o `--to-code=UTF-8`) establece el del archivo convertido. Todas las codificaciones soportadas por el comando `iconv` se listan con el comando `iconv -l` o `iconv --list`. En lugar de usar la redirección de la salida, como en el ejemplo, también se puede usar la opción `-o converted.txt` o `--output converted.txt`.

Ejercicios guiados

1. Basado en la siguiente salida del comando `date`, ¿cuál es la zona horaria del sistema en notación GMT?

```
$ date
Mon Oct 21 18:45:21 +05 2019
```

2. ¿A qué archivo debe apuntar el enlace simbólico `/etc/localtime` para que `Europa/Bruselas` sea la hora local por defecto del sistema?

3. Es posible que los caracteres de los archivos de texto no se representen correctamente en un sistema con una codificación de caracteres diferente de la utilizada en el documento de texto. ¿Cómo podría usarse `iconv` para convertir el archivo codificado WINDOWS-1252 `old.txt` en el archivo `new.txt` usando la codificación UTF-8?

Ejercicios de exploración

1. ¿Qué comando hará que `Pacific/Auckland` sea la zona horaria por defecto para la sesión de shell actual?

2. El comando `uptime` muestra, entre otras cosas, el promedio de carga del sistema en números fraccionarios. Utiliza la configuración actual de la región para decidir si el separador de decimales debe ser un punto o una coma. Si, por ejemplo, el locale actual está configurado como `de_DE.UTF-8` (el locale estándar de Alemania), `uptime` utilizará una coma como separador. Sabiendo que en el idioma inglés americano el punto se usa como separador, ¿qué comando hará que `uptime` muestre las fracciones usando un punto en lugar de una coma para el resto de la sesión actual?

3. El comando `iconv` reemplazará todos los caracteres fuera del conjunto con un signo de interrogación. Si se añade `//TRANSLIT` a la codificación de destino, los caracteres no representados en el conjunto de caracteres destino serán reemplazados (transliterados) por uno o más caracteres de aspecto similar. ¿Cómo podría usarse este método para convertir un archivo de texto UTF-8 llamado `readme.txt` a un archivo ASCII plano llamado `ascii.txt`?

Resumen

Esta lección cubre cómo configurar un sistema Linux para que funcione con idiomas y configuraciones de fecha y hora personalizados. También se tratan los conceptos y ajustes de codificación de caracteres, ya que son muy importantes para representar correctamente el contenido del texto. La lección abarca los siguientes temas:

- ¿Cómo los sistemas Linux seleccionan el idioma para mostrar los mensajes del shell?
- Entender cómo las zonas horarias afectan a la hora local.
- Cómo identificar la zona horaria apropiada y modificar la configuración del sistema en consecuencia.
- ¿Qué son las codificaciones de caracteres y cómo convertirlas?

Los comandos y procedimientos abordados fueron:

- Variables de entorno relacionadas con la región, fecha y hora, como `LC_ALL`, `LANG` y `TZ`.
- `/etc/timezone`
- `/etc/localtime`
- `/usr/share/zoneinfo/`
- `locale`
- `tzselect`
- `timedatectl`
- `date`
- `iconv`

Respuestas a los ejercicios guiados

1. Basado en la siguiente salida del comando `date`, ¿cuál es la zona horaria del sistema en notación GMT?

   ```
   $ date
   Mon Oct 21 18:45:21 +05 2019
   ```

   Es la zona horaria `Etc/GMT+5`.

2. ¿A qué archivo debe apuntar el enlace simbólico `/etc/localtime` para que `Europa/Bruselas` sea la hora local por defecto del sistema?

   El enlace `/etc/localtime` debe apuntar a `/usr/share/zoneinfo/Europe/Brussels`.

3. Es posible que los caracteres de los archivos de texto no se representen correctamente en un sistema con una codificación de caracteres diferente de la utilizada en el documento de texto. ¿Cómo podría usarse `iconv` para convertir el archivo codificado WINDOWS-1252 `old.txt` en el archivo `new.txt` usando la codificación UTF-8?

   El comando `iconv -f WINDOWS-1252 -t UTF-8 -o new.txt old.txt` realizará la conversión deseada.

Respuestas a los ejercicios de exploración

1. ¿Qué comando hará que `Pacific/Auckland` sea la zona horaria por defecto para la sesión de shell actual?

   `export TZ=Pacific/Auckland`

2. El comando `uptime` muestra, entre otras cosas, el promedio de carga del sistema en números fraccionarios. Utiliza la configuración actual de la región para decidir si el separador de decimales debe ser un punto o una coma. Si, por ejemplo, el locale actual está configurado como `de_DE.UTF-8` (el locale estándar de Alemania), `uptime` utilizará una coma como separador. Sabiendo que en el idioma inglés americano el punto se usa como separador, ¿qué comando hará que `uptime` muestre las fracciones usando un punto en lugar de una coma para el resto de la sesión actual?

   El comando `export LC_NUMERIC=en_US.UTF-8` o `export LC_ALL=en_US.UTF-8`.

3. El comando `iconv` reemplazará todos los caracteres fuera del conjunto con un signo de interrogación. Si se añade `//TRANSLIT` a la codificación de destino, los caracteres no representados en el conjunto de caracteres destino serán reemplazados (transliterados) por uno o más caracteres de aspecto similar. ¿Cómo podría usarse este método para convertir un archivo de texto UTF-8 llamado `readme.txt` a un archivo ASCII plano llamado `ascii.txt`?

   El comando `iconv -f UTF-8 -t ASCII//TRANSLIT -o ascii.txt readme.txt` realizará la conversión deseada.



Tema 108: Servicios esenciales del sistema



108.1 Mantener la hora del sistema

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 108.1

Importancia

3

Áreas de conocimiento clave

- Ajustar la fecha y hora del sistema.
- Ajustar el reloj de hardware a la hora correcta en UTC.
- Configuración de la zona horaria correcta.
- Configuración NTP básica usando ntpd y chrony.
- Saber cómo usar el servicio pool.ntp.org.
- Conocer el comando ntpq.

Lista parcial de archivos, términos y utilidades

- `/usr/share/zoneinfo/`
- `/etc/timezone`
- `/etc/localtime`
- `/etc/ntp.conf`
- `/etc/chrony.conf`
- `date`
- `hwclock`

- `timedatectl`
- `ntpd`
- `ntpdate`
- `chronyc`
- `pool.ntp.org`



108.1 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	108 Servicios esenciales del sistema
Objetivo:	108.1 Mantener la hora del sistema
Lección:	1 de 2

Introducción

Mantener la exactitud de la hora es absolutamente crucial para la informática moderna, sin embargo,su implementación es sorprendentemente compleja.. La práctica de mantener la hora parece trivial para un usuario final, pero el sistema necesita ser capaz de manejar muchas idiosincrasias y casos extremos de forma inteligente. Hay que tener en cuenta que las zonas horarias no son estáticas, sino que pueden modificarse por una decisión administrativa o política. Un país puede optar por dejar de utilizar el horario de verano. Cualquier programa debe ser capaz de manejar esos cambios de forma lógica. Afortunadamente para los administradores de sistemas, las soluciones para el control de la hora en el sistema operativo Linux son maduras, robustas y generalmente funcionan sin mucha interferencia. Cuando un equipo informático Linux arranca, empieza a mantener el tiempo. Nos referimos a esto como un *reloj del sistema*, ya que es actualizado por el sistema operativo. Además, los ordenadores modernos también tendrán un *reloj de hardware* o *de tiempo real*. Este reloj de hardware es a menudo una característica de la placa madre y matiene la hora independientemente de si el ordenador está funcionando o no. Durante el arranque, la hora del sistema se ajusta desde el reloj de hardware, pero en la mayoría de los casos estos dos relojes funcionan independientemente el uno del otro. En la mayoría de los sistemas modernos de Linux, la hora del sistema y del hardware están sincronizados con el *tiempo de la red*, que se implementa mediante el *Protocolo de Tiempo de Red* (NTP). En la gran mayoría de los casos, la única

configuración que un usuario normal tendrá que hacer es establecer su zona horaria y el NTP se encargará del resto. Sin embargo, cubriremos algunas formas de trabajar con la hora manualmente y los detalles de la configuración de la hora de red se tratarán en la próxima lección. === Local vs TIempo Universal El reloj del sistema está ajustado al Tiempo Universal Coordinado (UTC), que es la hora local de Greenwich, Reino Unido. Normalmente un usuario quiere saber su `hora local`. La hora local se calcula tomando la hora UTC y aplicando un *offset* basado en la zona horaria y en un "Horario de verano" De esta manera, se puede evitar mucha complejidad. El reloj del sistema puede ajustarse a la hora UTC o a la hora local, pero se recomienda que se ajuste a la hora UTC.

Date

El comando `date` es una utilidad básica que simplemente imprime la hora local:

```
$ date
Sun Nov 17 12:55:06 EST 2019
```

Modificar las opciones del comando `date` cambiará el formato de la salida.

Por ejemplo, un usuario puede usar `date -u` para ver la hora UTC actual.

```
$ date -u
Sun Nov 17 18:02:51 UTC 2019
```

Algunas otras opciones de uso común devolverán la hora local a un formato que se adhiere a un modelo RFC aceptado:

`-I`

Fecha/hora en formato ISO 8601. Si se añade `date` (`-Idate`) se limitará la salida a la fecha solamente. Otros formatos son `hours` para horas, `minutes` para minutos, `seconds` para segundos y `ns` para nanosegundos.

`-R`

Devuelve la fecha y la hora en formato RFC 5322.

`--rfc-3339`

Devuelve la fecha y la hora en formato RFC 3339.

El formato de `date` puede ser personalizado por el usuario con secuencias especificadas en la página de manual. Por ejemplo, la hora actual puede ser formateada como la hora de Unix de esta manera:

```
$ date +%s
1574014515
```

En la página del manual de `date` podemos ver que `%s` se refiere al tiempo de Unix.

El tiempo Unix se utiliza internamente en la mayoría de los sistemas tipo Unix. Almacena la hora UTC como el número de segundos desde *Epoch*, que ha sido definido como el 1 de enero de 1970.

NOTE El número de bits necesarios para almacenar el tiempo de Unix en el presente es de 32 bits. Hay un problema futuro en el que 32 bits serán insuficientes para contener la hora actual en formato Unix. Esto causará serios problemas para cualquier sistema Linux de 32 bits. Afortunadamente, esto no ocurriría sino hasta el 19 de enero de 2038.

Utilizando estas secuencias, somos capaces de dar formato a la fecha y la hora en casi cualquier formato requerido por cualquier aplicación. Por supuesto, en la mayoría de los casos es preferible atenerse a una norma aceptada.

Además, `date --date` puede usarse para dar formato a una hora que no es la actual. En este escenario, un usuario puede especificar la fecha que se aplicará al sistema utilizando la hora de Unix, por ejemplo:

```
$ date --date='@1564013011'
Wed Jul 24 20:03:31 EDT 2019
```

Usar la opción `--debug` puede ser muy útil para asegurar que una fecha pueda ser analizada con éxito. Observe lo que sucede cuando se pasa una fecha válida al comando:

```
$ date --debug --date="Fri, 03 Jan 2020 14:00:17 -0500"
date: parsed day part: Fri (day ordinal=0 number=5)
date: parsed date part: (Y-M-D) 2020-01-03
date: parsed time part: 14:00:17 UTC-05
date: input timezone: parsed date/time string (-05)
date: using specified time as starting value: '14:00:17'
date: warning: day (Fri) ignored when explicit dates are given
date: starting date/time: '(Y-M-D) 2020-01-03 14:00:17 TZ=-05'
date: '(Y-M-D) 2020-01-03 14:00:17 TZ=-05' = 1578078017 epoch-seconds
date: timezone: system default
date: final: 1578078017.000000000 (epoch-seconds)
date: final: (Y-M-D) 2020-01-03 19:00:17 (UTC)
date: final: (Y-M-D) 2020-01-03 14:00:17 (UTC-05)
```

Esta puede ser una herramienta útil cuando se trata de resolver problemas con una aplicación que genera una fecha.

Reloj de hardware

Un usuario puede ejecutar el comando `hclock` para ver como la hora se mantiene en el reloj en tiempo real. Este comando requerirá privilegios elevados, por lo que en este caso usaremos `sudo` para ejecutar al comando:

```
$ sudo hwclock
2019-11-20 11:31:29.217627-05:00
```

Usando la opción `--verbose` la salida del comando tendrá más detalles que pueden ser útiles para la resolución de problemas:

```
$ sudo hwclock --verbose
hwclock from util-linux 2.34
System Time: 1578079387.976029
Trying to open: /dev/rtc0
Using the rtc interface to the clock.
Assuming hardware clock is kept in UTC time.
Waiting for clock tick...
...got clock tick
Time read from Hardware Clock: 2020/01/03 19:23:08
Hw clock time : 2020/01/03 19:23:08 = 1578079388 seconds since 1969
Time since last adjustment is 1578079388 seconds
Calculated Hardware Clock drift is 0.000000 seconds
2020-01-03 14:23:07.948436-05:00
```

Fíjese en la línea que contiene `Calculated Hardware Clock drift`. Esta salida puede decirle si la hora del sistema y la del hardware se desvían una de otra.

timedatectl

`timedatectl` es un comando que puede utilizarse para comprobar el estado general de la hora y la fecha, incluyendo si la hora de red se ha sincronizado o no (dicho protocolo se tratara en la próxima lección.

Por defecto `timedatectl` devuelve información similar a `date`, pero con la adición de la hora RTC (hardware) así como el estado del servicio NTP:

```
$ timedatectl
               Local time: Thu 2019-12-05 11:08:05 EST
           Universal time: Thu 2019-12-05 16:08:05 UTC
                 RTC time: Thu 2019-12-05 16:08:05
                Time zone: America/Toronto (EST, -0500)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

Ajustar el tiempo utilizando timedatectl

No hay un servidor NTP disponible, se recomienda usar `timedatectl` en lugar de `date` o `hwclock` para fijar la hora:


```
# timedatectl set-time '2011-11-25 14:00:00'
```

El proceso es similar a `date`. El usuario también puede establecer la hora independientemente de la fecha usando el formato HH:MM:SS.

Ajustar la zona horaria utilizando timedatectl

`timedatectl` es la forma preferida de establecer la zona horaria local en sistemas Linux basados en `Systemd` cuando no existe una interfaz gráfica. `Timedatectl` listará posibles zonas horarias y luego la zona horaria puede ser establecida usando una de estas como argumento.

Primero haremos una lista de posibles zonas horarias:

```
$ timedatectl list-timezones
Africa/Abidjan
Africa/Accra
Africa/Algiers
Africa/Bissau
Africa/Cairo
...
```

La lista de posibles zonas horarias es larga, por lo que en este caso se recomienda el uso del comando `grep`.

A continuación podemos establecer la zona horaria usando uno de los elementos de la lista que fue devuelto:

```
$ timedatectl set-timezone Africa/Cairo
$ timedatectl
               Local time: Thu 2019-12-05 18:18:10 EET
           Universal time: Thu 2019-12-05 16:18:10 UTC
                 RTC time: Thu 2019-12-05 16:18:10
                Time zone: Africa/Cairo (EET, +0200)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

Tenga en cuenta que el nombre de la zona horaria debe ser exacto. `Africa/Cairo` por ejemplo cambiará la zona horaria, pero `cairo` o `africa/cairo` no lo hará.

Desactivar NTP usando timedatectl

En algunos casos podría ser necesario desactivar NTP. Esto podría hacerse usando `systemctl` pero lo demostraremos usando `timedatectl`:

```
# timedatectl set-ntp no
$ timedatectl
               Local time: Thu 2019-12-05 18:19:04 EET Universal time: Thu 2019-12-05
16:19:04 UTC
                 RTC time: Thu 2019-12-05 16:19:04
                Time zone: Africa/Cairo (EET, +0200)
              NTP enabled: no
         NTP synchronized: no
          RTC in local TZ: no
               DST active: n/a
```

Establecer la zona horaria sin timedatectl

La configuración de la información de la zona horaria es un paso estándar cuando se instala Linux en una nueva máquina. Si hay un proceso de instalación gráfico, lo más probable es que se maneje sin ninguna otra entrada del usuario.

El directorio `/usr/share/zoneinfo` contiene información de las diferentes zonas horarias posibles. En el directorio `zoneinfo`, hay subdirectorios que contienen los nombres de los continentes así como otros enlaces simbólicos. Se recomienda encontrar el `zoneinfo` de su región a partir de su continente.

Los archivos `zoneinfo` contienen las reglas necesarias para calcular el desfase de la hora local en relación con UTC, y también son importantes si su región hace uso del horario de verano. El contenido de `/etc/localtime` será leído cuando Linux necesite determinar la zona horaria local. Para establecer la zona horaria sin el uso de una interfaz gráfica, el usuario debe crear un enlace simbólico para su ubicación desde `/usr/share/zoneinfo` a `/etc/localtime`. Por ejemplo:

```
$ ln -s /usr/share/zoneinfo/Canada/Eastern /etc/localtime
```

Después de establecer la zona horaria correcta, se recomienda ejecutar:

```
# hwclock --systohc
```

Esto ajustará el *reloj de hardware* desde el *reloj de sistema* (es decir, el reloj de tiempo real se ajustará a la misma hora que `date`). Tenga en cuenta que este comando se ejecuta con privilegios de root, en

este caso se ejecutó desde una session de root.

`/etc/timezone` es similar a `/etc/localtime`. Es una representación de datos de la zona horaria local, y como tal puede ser leída usando `cat`:

```
$ cat /etc/timezone
America/Toronto
```

Tenga en cuenta que este archivo no es utilizado por todas las distribuciones de Linux.

Establecer la fecha y la hora sin timedatectl

NOTE La mayoría de los sistemas modernos de Linux usan `systemd` para su configuración y servicios, y como tal no se recomienda usar `date` o `hwclock` para fijar la hora. `Systemd` usa `timedatectl` para esto. No obstante, es importante conocer estos comandos heredados en caso de que tenga que administrar un sistema antiguo.

Utilizando date

`date` tiene una opción para ajustar la hora del sistema. Estas son: `--set` o `-s` para fijar la fecha y la hora. También puede usar `--debug` para verificar la sintaxis correcta correcto del comando:

```
# date --set="11 Nov 2011 11:11:11"
```

Tenga en cuenta que se requieren privilegios de root para fijar la fecha aquí. También podemos cambiar la hora o la fecha de forma independiente:

```
# date +%Y%m%d -s "20111125"
```

Aquí debemos especificar las secuencias para que nuestra cadena sea interpretada correctamente. Por ejemplo, `%Y` se refiere al año, y así los primeros cuatro dígitos `2011` se interpretarán como el año 2011. De manera similar, `%T` es la secuencia para el tiempo, y se demuestra así:

```
# date +%T -s "13:11:00"
```

Después de cambiar la hora del sistema, se recomienda también ajustar el reloj del hardware para que ambos relojes, el del sistema y el del hardware, estén sincronizados:

```
# hwclock --systohc
```

`systohc` significa “system clock to hardware clock”.

Utilizando hwclock

En lugar de ajustar el reloj del sistema y actualizar el reloj del hardware, puede optar por invertir el proceso. Empezaremos por ajustar el reloj del hardware:

```
# hwclock --set --date "4/12/2019 11:15:19"
# hwclock
Fri 12 Apr 2019 6:15:19 AM EST -0.562862 seconds
```

Note que por defecto el `hwclock` espera la hora UTC, pero devuelve la hora local por defecto.

Después de ajustar el reloj del hardware, tendremos que actualizar el reloj del sistema a partir de este. `hctosys` significa “hardware clock to system clock”.

```
# hwclock --hctosys
```


Ejercicios guiados

1. Indique si los siguientes comandos están mostrando o modificando la *hora del sistema* o la *hora del hardware*:

Comando(s)	Sistema	Hardware	Ambos
`date -u`			
`hwclock --set --date "12:00:00"`			
`timedatectl`			
`timedatectl | grep RTC`			
`hwclock --hctosys`			
`date +%T -s "08:00:00"`			
`timedatectl set-time 1980-01-10`			

2. Observe la siguiente salida, y luego corrija el formato del argumento para que el comando sea exitoso:

```
$ date --debug --date "20/20/12 0:10 -3"

date: warning: value 20 has less than 4 digits. Assuming MM/DD/YY[YY]
date: parsed date part: (Y-M-D) 0002-20-20
date: parsed time part: 00:10:00 UTC-03
date: input timezone: parsed date/time string (-03)
date: using specified time as starting value: '00:10:00'
date: error: invalid date/time value:
date:     user provided time: '(Y-M-D) 0002-20-20 00:10:00 TZ=-03'
date:        normalized time: '(Y-M-D) 0003-08-20 00:10:00 TZ=-03'
date:                                  ---- --
date:      possible reasons:
date:        numeric values overflow;
date:        incorrect timezone
date: invalid date '20/20/2 0:10 -3'
```

3. Use el comando `date` y las secuencias para que el mes del sistema sea febrero. Deje el resto de la fecha y la hora sin cambios.

4. Asumiendo que el comando anterior tuvo éxito, use `hwclock` para ajustar el reloj del hardware desde el reloj del sistema.

5. Hay un lugar llamado `eucla`. ¿De qué continente forma parte? Use el comando `grep` para averiguarlo.

6. Establezca su zona horaria actual en la de `eucla`.

Ejercicios de exploración

1. ¿Qué método de ajuste de tiempo es el óptimo? ¿En qué escenario podría ser imposible el método preferido?

2. ¿Por qué cree que hay tantos métodos para lograr lo mismo, es decir, establecer la fecha y hora del sistema?

3. Después del 19 de enero de 2038, Linux System Time requerirá un número de 64 bits para almacenar. Sin embargo, es posible que podamos elegir simplemente establecer un "nuevo epoch". Por ejemplo, el 1 de enero de 2038 a medianoche podría establecerse una nueva época de `0`. ¿Por qué cree que esto no se ha convertido en la solución preferida?

Resumen

En esta lección aprendió:

- A mostrar la hora en diferentes formatos desde la línea de comandos.
- La diferencia entre el reloj del sistema y el reloj del hardware en Linux.
- Ajustar manualmente el reloj del sistema.
- Establecer manualmente el reloj del hardware.
- Cambiar la zona horaria del sistema.

Comandos usados en esta lección:

`date`
: Visualizar o cambiar el reloj del sistema. Otras opciones:

  `-u`
  : Muestra la hora UTC.

  `+%s`
  : Usa una secuencia para mostrar el tiempo de la época.

  `--date=`
  : Establece una hora específica para mostrar, en lugar de la hora actual.

  `--debug`
  : Muestra mensajes de depuración al interpretar una fecha introducida por el usuario.

  `-s`
  : Ajusta el reloj del sistema manualmente.

`hwclock`
: Muestra o cambia el reloj del hardware.

  `--systohc`
  : Usa el reloj del sistema para ajustar el reloj del hardware.

  `--hctosys`
  : Usa el reloj de hardware para ajustar el reloj del sistema.

`--set --date`
: Ajusta el reloj de hardware manualmente.

`timedatectl`
: Muestra los relojes del sistema y del hardware, así como la configuración NTP en los sistemas Linux basados en Systemd.

`set-time`
: Ajusta la hora manualmente.

`list-timezones`
: Lista las posibles zonas horarias.

`set-timezone`
: Configura la zona horaria manualmente.

`set-ntp`
: Activa/desactiva NTP.

Respuesta a los ejercicios guiados

1. Indique si los siguientes comandos están mostrando o modificando la *hora del sistema* o la *hora del hardware*:

Comando(s)	Sistema	Hardware	Ambos
`date -u`	X		
`hwclock --set --date "12:00:00"`		X	
`timedatectl`			X
`timedatectl | grep RTC`		X	
`hwclock --hctosys`	X		
`date +%T -s "08:00:00"`	X		
`timedatectl set-time 1980-01-10`			X

2. Observe la siguiente salida, y luego corrija el formato del argumento para que el comando sea exitoso:

```
$ date --debug --date "20/20/12 0:10 -3"

date: warning: value 20 has less than 4 digits. Assuming MM/DD/YY[YY]
date: parsed date part: (Y-M-D) 0002-20-20
date: parsed time part: 00:10:00 UTC-03
date: input timezone: parsed date/time string (-03)
date: using specified time as starting value: '00:10:00'
date: error: invalid date/time value:
date:     user provided time: '(Y-M-D) 0002-20-20 00:10:00 TZ=-03'
date:       normalized time: '(Y-M-D) 0003-08-20 00:10:00 TZ=-03'
date:                                  ---- --
date:      possible reasons:
date:        numeric values overflow;
date:        incorrect timezone
date: invalid date ‘20/20/2 0:10 -3’
```

`date --debug --set "12/20/20 0:10 -3"`

3. Use el comando `date` y las secuencias para que el mes del sistema sea febrero. Deje el resto de la

fecha y la hora sin cambios.

```
date +%m -s "2"
```

4. Asumiendo que el comando anterior tuvo éxito, use `hwclock` para ajustar el reloj del hardware desde el reloj del sistema.

```
hwclock -systohc
```

5. Hay un lugar llamado `eucla`. ¿De qué continente forma parte? Usa el comando `grep` para averiguarlo.

```
timedatectl list-timezones \| grep -i eucla
```

o

```
grep -ri eucla /usr/share/zoneinfo
```

6. Establezca su zona horaria actual en la de `eucla`.

```
timedatectl set-timezone 'Australia/Eucla'
```

o

```
ln -s /usr/share/zoneinfo/Australia/Eucla /etc/localtime
```

Respuestas a los ejercicios de exploración

1. ¿Qué método de ajuste de tiempo es el óptimo? ¿En qué escenario podría ser imposible el método preferido?

   En la mayoría de las distribuciones de Linux, el NTP está habilitado por defecto y debería dejarse que establezca la hora del sistema sin interferencias. Sin embargo, si hay un sistema Linux que no está conectado a Internet, NTP será inaccesible. Por ejemplo, un sistema Linux integrado que funcione en un equipo industrial podría no tener conectividad a la red.

2. ¿Por qué crees que hay tantos métodos para lograr lo mismo, es decir, establecer la fecha y hora del sistema?

   Dado que el establecimiento del tiempo ha sido un requisito de todos los sistemas *nix durante décadas, hay muchos métodos heredados para establecer el tiempo que aún se mantienen.

3. Después del 19 de enero de 2038, Linux System Time requerirá un número de 64 bits para almacenar. Sin embargo, es posible que podamos elegir simplemente establecer un “nuevo epoch”. Por ejemplo, el 1 de enero de 2038 a medianoche podría establecerse un nuevo epoch de `0`. ¿Por qué crees que esto no se ha convertido en la solución preferida?

   Para el 2038 la gran mayoría de las computadoras ya estarán funcionando con CPU de 64 bits, y el uso de un número de 64 bits no degradará el rendimiento de manera significativa. Sin embargo, sería imposible estimar los riesgos de “resetear” epoch de tal manera. Hay mucho software antiguo que podría ser afectado. Los bancos y las grandes empresas, por ejemplo, a menudo tienen una gran cantidad de programas antiguos de los que dependen para su uso interno. Así que este escenario, como muchos otros, es un estudio de las compensaciones. Cualquier sistema de 32 bits que siga funcionando en 2038 se vería afectado por un desbordamiento de "Epoch Time", pero el software heredado se vería afectado por el cambio de valor de Epoch.



108.1 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	108 Servicios esenciales del sistema
Objetivo:	108.1 Mantener la hora del sistema
Lección:	2 de 2

Introducción

Mientras que las computadoras personales son capaces de mantener una hora razonablemente precisa, la informática de producción y los entornos de red requieren que se mantenga una hora muy precisa. La hora más precisa se mide con *relojes de referencia*, que suelen ser relojes atómicos. El mundo moderno ha ideado un sistema en el que todos los dispositivos informáticos conectados a Internet pueden sincronizarse con estos relojes utilizando lo que se conoce como el *Protocolo de Tiempo de Red* (NTP). Un sistema informático con NTP podrá sincronizar sus relojes de sistema con la hora proporcionada por los relojes de referencia. Si la hora del sistema y la hora medida en estos servidores son diferentes, entonces el ordenador adelentará o atrasará su hora interna del sistema de forma gradual hasta que la hora coincida con la de red.

NTP utiliza una estructura jerárquica para distribuir la hora. Los relojes de referencia están conectados a servidores situados en la parte superior de la jerarquía. Estos servidores son máquinas de *estrato 1* y normalmente no son accesibles al público. Sin embargo, las máquinas del estrato 1 son accesibles por las máquinas del estrato 2, estas a su vez son accesibles a las de estrato 3, y así sucesivamente. Los servidores de estrato 2 son accesibles al público, al igual que las máquinas de menor jerarquía. Cuando se configura NTP para una red grande, es una buena práctica tener un pequeño número de ordenadores conectados a los servidores de estrato 2+, y luego hacer que esas

máquinas proporcionen NTP a todas las demás máquinas. De esta manera, se pueden minimizar las demandas sobre las máquinas de estrato 2.

Hay algunos términos importantes que surgen cuando se habla de NTP. Algunos de estos términos se usan en los comandos que implementaremos para comprobar el estado de NTP en nuestras máquinas:

Offset
: Se refiere a la diferencia absoluta entre la hora del sistema y la hora NTP. Por ejemplo, si el reloj del sistema marca las 12:00:02 y la hora NTP marca las 11:59:58, el desfase entre los dos relojes es de cuatro segundos.

Step
: Si el desfase de horario entre el proveedor NTP y un consumidor es superior a 128ms, entonces NTP realizará un único cambio significativo en la hora del sistema, en lugar de atrasar o adelantar la hora del sistema. Esto se llama *stepping*.

Slew
: Se refiere a los cambios realizados en la hora del sistema cuando el offset entre la hora del sistema y la NTP es inferior a 128 ms. Si este es el caso, los cambios se harán gradualmente. Esto se conoce como *slewing*.

Insane Time
: Si el offset entre la hora del sistema y la hora NTP es superior a 17 minutos, la hora del sistema se considera *insane* y el demonio NTP no introducirá ningún cambio en la hora del sistema. Habrá que tomar medidas especiales para que la hora del sistema esté dentro de los 17 minutos de la hora correcta.

Drift
: Se refiere al fenómeno por el que dos relojes se desincronizan con el tiempo. Esencialmente, si dos relojes están inicialmente sincronizados pero luego se desincronizan con el tiempo, entonces se está produciendo una deriva del reloj.

Jitter
: La fluctuación se refiere a la cantidad de desviación desde la última vez que se consultó un reloj. Así, si la última sincronización NTP se produjo hace 17 minutos, y el desfase entre el proveedor y el consumidor NTP es de 3 milisegundos, entonces 3 milisegundos es el jitter.

Ahora discutiremos algunas de las formas específicas en que Linux implementa NTP.

timedatectl

Si su distribución de Linux utiliza `timedatectl`, entonces por defecto implementa un cliente *SNTP* en lugar de una implementación completa de NTP. Esta es una implementación menos compleja de la hora de red y significa que su máquina no servirá NTP a otros ordenadores conectados.

En este caso, SNTP no funcionará a menos que el servicio `timesyncd` se esté ejecutando. Como con todos los servicios systemd, podemos verificar que se está corriendo con:

```
$ systemctl status systemd-timesyncd
● systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled; vendor
preset: enabled)
  Drop-In: /lib/systemd/system/systemd-timesyncd.service.d
           └─disable-with-time-daemon.conf
   Active: active (running) since Thu 2020-01-09 21:01:50 EST; 2 weeks 1 days ago
     Docs: man:systemd-timesyncd.service(8)
 Main PID: 1032 (systemd-timesyn)
   Status: "Synchronized to time server for the first time 91.189.89.198:123
(ntp.ubuntu.com)."
    Tasks: 2 (limit: 4915)
   Memory: 3.0M
   CGroup: /system.slice/systemd-timesyncd.service
           └─1032 /lib/systemd/systemd-timesyncd

Jan 11 13:06:18 NeoMex systemd-timesyncd[1032]: Synchronized to time server for the
first time 91.189.91.157:123 (ntp.ubuntu.com).
...
```

El estado de la sincronización SNTP de `timedatectl` puede verificarse con `show-timesync`:

```
$ timedatectl show-timesync --all
LinkNTPServers=
SystemNTPServers=
FallbackNTPServers=ntp.ubuntu.com
ServerName=ntp.ubuntu.com
ServerAddress=91.189.89.198
RootDistanceMaxUSec=5s
PollIntervalMinUSec=32s
PollIntervalMaxUSec=34min 8s
PollIntervalUSec=34min 8s
NTPMessage={ Leap=0, Version=4, Mode=4, Stratum=2, Precision=-23,
RootDelay=8.270ms, RootDispersion=18.432ms, Reference=91EECB0E,
OriginateTimestamp=Sat 2020-01-25 18:35:49 EST, ReceiveTimestamp=Sat 2020-01-25
18:35:49 EST, TransmitTimestamp=Sat 2020-01-25 18:35:49 EST,
DestinationTimestamp=Sat 2020-01-25 18:35:49 EST, Ignored=no PacketCount=263,
Jitter=2.751ms }
Frequency=-211336
```

Esta configuración puede ser adecuada para la mayoría de las situaciones, pero como se ha señalado anteriormente, será insuficiente si se espera sincronizar varios clientes en una red. En este caso se recomienda instalar un cliente NTP completo.

NTP Daemon

La hora del sistema se compara con la hora de la red en un horario regular. Para que esto funcione debemos tener un *daemon* que se ejecute en segundo plano. En muchos sistemas Linux, el nombre de este demonio es `ntpd`. Este permitirá a una máquina no sólo ser un *consumidor de tiempo* (es decir, capaz de sincronizar su propio reloj desde una fuente externa), sino también *proveer* el tiempo a otras máquinas.

Supongamos que nuestro ordenador está basado en systemd y que utiliza `systemctl` para controlar los demonios. Instalaremos los paquetes `ntp` utilizando el gestor de paquetes apropiado y luego nos aseguraremos de que nuestro demonio `ntpd` se está ejecutando comprobando su estado:


```
$ systemctl status ntpd

  ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset:
disabled)
   Active: active (running) since Fri 2019-12-06 03:27:21 EST; 7h ago
  Process: 856 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited,
status=0/SUCCESS)
 Main PID: 867 (ntpd)
   CGroup: /system.slice/ntpd.service
           `-867 /usr/sbin/ntpd -u ntp:ntp -g
```

En algunos casos puede ser necesario iniciar y habilitar `ntpd`. En la mayoría de las máquinas Linux esto se logra con:

```
# systemctl enable ntpd && systemctl start ntpd
```

Las consultas NTP se realizan en el puerto TCP 123. Si NTP falla, asegúrese de que este puerto está abierto y a la escucha.

Configuración NTP

NTP es capaz de sondear varias fuentes y seleccionar las mejores candidatas para utilizarlas en el ajuste de la hora del sistema. Si se pierde una conexión de red, NTP utiliza los ajustes anteriores de su historial para estimar los ajustes futuros. Dependiendo de su distribución de Linux, la lista de servidores de tiempo de red se almacenará en diferentes lugares. Supongamos que `ntp` está instalado en su máquina.

El archivo `/etc/ntp.conf` contiene información de configuración sobre cómo su sistema se sincroniza con la hora de la red. Este archivo puede ser leído y modificado usando cualquier editor de texto plano.

Por defecto, los servidores NTP utilizados se especificarán en una sección como ésta:

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

La sintaxis para añadir servidores NTP es la siguiente:

```
server (IP Address)
server server.url.localhost
```

Las direcciones de los servidores pueden ser direcciones IP o URL si se ha configurado correctamente el DNS. En este caso, siempre se consultará el servidor. Un administrador de red también podría considerar el uso (o la creación) de un *pool*. En este caso, suponemos que hay varios proveedores NTP, todos ellos ejecutando demonios NTP y con la misma hora. Cuando un cliente consulta un pool, se selecciona un proveedor al azar. Esto ayuda a distribuir la carga de la red entre muchas máquinas para que ninguna máquina del pool esté manejando todas las consultas NTP. Comúnmente, `/etc/ntp.conf` se cargará con un pool de servidores llamado `pool.ntp.org`. Así, por ejemplo, `servidor 0.centos.pool.ntp.org` es un pool NTP por defecto proporcionado a las máquinas CentOS.

pool.ntp.org

Los servidores NTP utilizados por defecto son un proyecto de código abierto. Puede encontrar más información en ntppool.org.

> Considere si el Pool NTP es apropiado para su uso. Si el negocio, la organización o la vida humana depende de tener la hora correcta o puede ser perjudicada por estar incorrecta, no debería "simplemente sacarla de Internet". El NTP Pool es generalmente de muy alta calidad, pero es un servicio dirigido por voluntarios en su tiempo libre. Por favor, hable con sus proveedores de equipos y servicios para conseguir un servicio local y fiable para usted. Consulte también nuestras condiciones de servicio. Recomendamos los servidores de tiempo de Meinberg, pero también puede encontrar servidores de tiempo de End Run, Spectracom y muchos otros.
>
> — ntppool.org

ntpdate

Durante la configuración inicial, la hora del sistema y la de NTP pueden estar muy desincronizadas. Si el *offset* entre la hora del sistema y la del NTP es superior a 17 minutos, el demonio NTP no realizará cambios en la hora del sistema. En este caso será necesaria la intervención manual.

En primer lugar, si `ntpd` se está ejecutando será necesario *detener* el servicio. Utilice `systemctl stop ntpd` para hacerlo.

A continuación, utilice `ntpdate pool.ntp.org` para realizar una única sincronización inicial, donde

`pool.ntp.org` se refiere a la dirección IP o URL de un servidor NTP. Puede ser necesaria más de una sincronización.

ntpq

`ntpq` es una utilidad para monitorizar el estado de NTP. Una vez que el demonio NTP se ha iniciado y configurado, `ntpq` se podrá utilizar para comprobar su estado:

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+37.44.185.42    91.189.94.4      3 u   86  128  377  126.509  -20.398   6.838
+ntp2.0x00.lv    193.204.114.233  2 u   82  128  377  143.885   -8.105   8.478
*inspektor-vlan1 121.131.112.137  2 u   17  128  377  112.878  -23.619   7.959
 b1-66er.matrix. 18.26.4.105      2 u  484  128   10   34.907   -0.811  16.123
```

En este caso `-p` (*print*) imprimirá un resumen de los pares. Las direcciones de host también pueden ser devueltas como direcciones IP usando `-n`.

`remote`

nombre de host del proveedor NTP.

`refid`

ID de referencia del proveedor NTP.

`st`

Estrato del proveedor.

`when`

Número de segundos desde la última consulta.

`poll`

Número de segundos entre consultas.

`reach`

ID de estado para indicar si se ha alcanzado un servidor. Las conexiones exitosas aumentarán este número en 1.

`delay`

Tiempo en ms entre la consulta y la respuesta del servidor.

`offset`

Tiempo en ms entre la hora del sistema y la hora NTP.

`jitter`

Offset en ms entre la hora del sistema y la NTP en la última consulta.

`ntpq` también tiene un modo interactivo, al que se accede cuando se ejecuta sin opciones ni argumentos. La opción `?` devolverá una lista de comandos que `ntpq` reconocerá.

chrony

`chrony` es otra forma de implementar NTP. Se instala por defecto en algunos sistemas Linux, pero está disponible para su descarga en las principales distribuciones. `chronyd` es el demonio de chrony, y `chronyc` es la interfaz de línea de comandos. Puede ser necesario iniciar y habilitar `chronyd` antes de interactuar con `chronyc`.

Si la instalación de chrony tiene una configuración por defecto, el comando `chronyc tracking` proporcionará información sobre NTP y la hora del sistema:

```
$ chronyc tracking
Reference ID    : 3265FB3D (bras-vprn-toroon2638w-lp130-11-50-101-251-61.dsl.)
Stratum         : 3
Ref time (UTC)  : Thu Jan 09 19:18:35 2020
System time     : 0.000134029 seconds fast of NTP time
Last offset     : +0.000166506 seconds
RMS offset      : 0.000470712 seconds
Frequency       : 919.818 ppm slow
Residual freq   : +0.078 ppm
Skew            : 0.555 ppm
Root delay      : 0.006151616 seconds
Root dispersion : 0.010947504 seconds
Update interval : 129.8 seconds
Leap status     : Normal
```

Esta salida contiene mucha información, más de la que está disponible en otras implementaciones. `Reference ID`:: El ID de referencia y el nombre con el que el ordenador está actualmente sincronizado.

`Stratum`
: Número de saltos a un ordenador con un reloj de referencia conectado.

`Ref time`
: Es la hora UTC a la que se realizó la última medición de la fuente de referencia.

`System time`
: Retraso del reloj del sistema desde el servidor sincronizado.

`Last offset`
: Offset estimado de la última actualización del reloj.

`RMS offset`
: Promedio a largo plazo del valor de offset.

`Frequency`
: Se trata de la tasa en la que el reloj del sistema pudiera estar incorrecto si el cronyd no lo corrigiera. Se proporciona en ppm (partes por millón).

`Residual freq`
: Frecuencia residual que indica la diferencia entre las mediciones de la fuente de referencia y la frecuencia que se utiliza actualmente.

`Skew`
: Límite de error estimado de la frecuencia.

`Root delay`
: Total de los retrasos de la ruta de red hacia el ordenador del estrato, desde el que se está sincronizando el ordenador.

`Leap status`
: Es el estado de salto que puede tener uno de los siguientes valores: normal, insertar segundo, borrar segundo o no sincronizado.

También podemos ver información detallada sobre la última actualización NTP válida:

```
# chrony ntpdata
Remote address  : 172.105.97.111 (AC69616F)
Remote port     : 123
Local address   : 192.168.122.81 (C0A87A51)
Leap status     : Normal
Version         : 4
Mode            : Server
Stratum         : 2
Poll interval   : 6 (64 seconds)
Precision       : -25 (0.000000030 seconds)
Root delay      : 0.000381 seconds
Root dispersion : 0.000092 seconds
Reference ID    : 61B7CE58 ()
Reference time  : Mon Jan 13 21:50:03 2020
Offset          : +0.000491960 seconds
Peer delay      : 0.004312567 seconds
Peer dispersion : 0.000000068 seconds
Response time   : 0.000037078 seconds
Jitter asymmetry: +0.00
NTP tests       : 111 111 1111
Interleaved     : No
Authenticated   : No
TX timestamping : Daemon
RX timestamping : Kernel
Total TX        : 15
Total RX        : 15
Total valid RX  : 15
```

Por último, `chronyc sources` devolverá información sobre los servidores NTP que se utilizan para sincronizar la hora:

```
$ chronyc sources
210 Number of sources = 0
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
```

Por el momento, esta máquina no tiene fuentes configuradas. Podemos añadir fuentes desde `pool.ntp.org` abriendo el fichero de configuración de chrony. Este suele estar ubicado en `/etc/chrony.conf`. Cuando abrimos este archivo, deberíamos ver que algunos servidores están listados por defecto:

```
210 Number of sources = 0
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
# Most computers using chrony will send measurement requests to one or
# more 'NTP servers'.  You will probably find that your Internet Service
# Provider or company have one or more NTP servers that you can specify.
# Failing that, there are a lot of public NTP servers.  There is a list
# you can access at http://support.ntp.org/bin/view/Servers/WebHome or
# you can use servers from the 3.arch.pool.ntp.org project.

! server 0.arch.pool.ntp.org iburst iburst
! server 1.arch.pool.ntp.org iburst iburst
! server 2.arch.pool.ntp.org iburst iburst

! pool 3.arch.pool.ntp.org iburst
```

Estos servidores también nos servirán de guía para la sintaxis a la hora de introducir nuestros propios servidores. Sin embargo, en este caso simplemente eliminaremos los `!` al principio de cada línea, descomentando así estas líneas y utilizando los servidores por defecto del proyecto `pool.ntp.org`.

Además, en este archivo podemos elegir cambiar la configuración por defecto en cuanto a skew y drift, así como la ubicación del driftfile y keyfile.

En esta máquina, necesitamos hacer una gran corrección inicial del reloj. Optaremos por descomentar la siguiente línea:

```
! makestep 1.0 3
```

Después de realizar los cambios en el archivo de configuración, reinicie el servicio `chronyd` y utilice `chronyc makestep` para escalar manualmente el reloj del sistema:

```
# chronyc makestep
200 OK
```

Y a continuación, utilice el `chronyc tracking` como antes para verificar que los cambios se han producido.

Ejercicios guiados

1. Introduzca el término apropiado para cada definición:

Definición	Término
Un ordenador que compartirá la hora de la red con usted	
Distancia de un reloj de referencia, en saltos o pasos.	
Diferencia entre la hora del sistema y la hora de la red	
Diferencia entre la hora del sistema y la hora de red desde el último sondeo NTP	
Grupo de servidores que proporcionan la hora de la red y que comparten la carga entre ellos	

2. Especifique cuál de los comandos utilizaría para dar salida a los siguientes valores:

Valor	`chronyc tracking`	`timedatectl show-timesync --all`	`ntpq -pn`	`chrony ntpdata`	`chronyc sources`
Jitter					
Drift					
Interval of Poll					
Offset					
Stratum					
IP Address of Provider					
Root Delay					

3. Está configurando una red empresarial que consta de un servidor Linux y varios ordenadores de sobremesa Linux. El servidor tiene una dirección IP estática de 192.168.0.101. Usted decide que el servidor se conectará a `pool.ntp.org` y luego proporcionará la hora NTP a los ordenadores de sobremesa. Describa la configuración del servidor y de los ordenadores de sobremesa.

4. Una máquina Linux tiene la hora incorrecta. Describa los pasos que daría para solucionar el problema de NTP.

Ejercicios de exploración

1. Investigue las diferencias entre SNTP y NTP.

SNTP	NTP

2. ¿Por qué un administrador de sistemas podría elegir *no* utilizar `pool.ntp.org`?

3. ¿Cómo podría un administrador de sistemas decidir unirse o contribuir de alguna manera al proyecto `pool.ntp.org`?

Resumen

En esta lección aprendió:

- Qué es NTP y por qué es importante.
- Configurar el demonio NTP utilizando el proyecto `pool.ntp.org`.
- Usar `ntpq` para verificar la configuración NTP.
- Emplear `chrony` como un servicio NTP alternativo.

Comandos usados en esta lección:

`timedatectl show-timesync --all`
: Muestra la información SNTP si se utiliza `timedatectl`.

`ntpdate <address>`
: Realiza una actualización manual de pasos NTP de una sola vez.

`ntpq -p`
: Imprime un historial de sondeos recientes de NTP. `-n` sustituirá las URLs por direcciones IP.

`chronyc tracking`
: Muestra el estado de NTP si se utiliza chrony.

`chronyc ntpdata`
: Muestra información NTP sobre el último sondeo.

`chronyc sources`
: Muestra información sobre los proveedores NTP.

`chronyc makestep`
: Realiza una actualización manual del paso NTP de una sola vez si utiliza chrony.

Respuesta a los ejercicios guiados

1. Introduzca el término apropiado para cada definición:

Definición	Término
Un ordenador que compartirá la hora de la red con usted	Provider
Distancia de un reloj de referencia, en saltos o pasos.	Stratum
Diferencia entre la hora del sistema y la hora de la red	Offset
iferencia entre la hora del sistema y la hora de la red desde el último sondeo NTP	Jitter
Grupo de servidores que proporcionan la hora de la red y que comparten la carga entre ellos	Pool

2. Especifique cuál de los comandos utilizaría para dar salida a los siguientes valores:

Valor	`chronyc tracking`	`timedatectl show-timesync --all`	`ntpq -pn`	`chrony ntpdata`	`chronyc sources`
Jitter		X	X		
Drift					
Interval of Poll	X	X	X	X	X
Offset	X		X	X	
Stratum	X	X	X	X	X
IP Address of Provider		X	X	X	X
Root Delay	X			X	

3. Está configurando una red empresarial que consta de un servidor Linux y varios ordenadores de sobremesa Linux. El servidor tiene una dirección IP estática de 192.168.0.101. Usted decide que el servidor se conectará a `pool.ntp.org` y luego proporcionará la hora NTP a los ordenadores de sobremesa. Describa la configuración del servidor y de los ordenadores de sobremesa.

   Asegúrese de que el servidor tiene un servicio ntpd en ejecución, en lugar de SNTP. Utilice los pools de `pool.ntp.org` en el archivo `/etc/ntp.conf` o `/etc/chrony.conf`. Para cada cliente,

especifique `192.168.0.101` en cada archivo `/etc/ntp.conf` o `/etc/chrony.conf`.

4. Una máquina Linux tiene la hora incorrecta. Describa los pasos que daría para solucionar el problema de NTP.

   En primer lugar, asegúrese de que la máquina está conectada a Internet. Utilice `ping` para ello. Compruebe que el servicio ntpd o SNTP se está ejecutando utilizando `systemctl status ntpd` o `systemctl status systemd-timesyncd`. Es posible que aparezcan mensajes de error que proporcionen información útil. Por último, utilice un comando como `ntpq -p` o `chrony tracking` para verificar si se ha realizado alguna solicitud. Si la hora del sistema es drásticamente diferente de la hora de la red, puede ser que la hora del sistema se considere "insane" y no se cambie sin intervención manual. En este caso, utilice un comando de la lección anterior o un comando como `ntpdate pool.ntp.org` para realizar una única sincronización ntp.

Respuestas a los ejercicios de exploración

1. Investigue las diferencias entre SNTP y NTP.

SNTP	NTP
Menos precisa	Más precisa.
Requiere menos recursos	Requiere más recursos.
No puede actuar como proveedor de tiempo	Puede actuar como proveedor de tiempo.
Sólo puede medir el tiempo por pasos (steps time)	Puede medir el tiempo por pasos o por pasos (slews time)
Solicita la hora a una sola fuente	Puede controlar varios servidores NTP y utilizar el proveedor óptimo.

2. ¿Por qué un administrador de sistemas podría elegir *no* utilizar `pool.ntp.org`?

   Desde ntppool.org: Si es absolutamente crucial tener la hora correcta, debería considerar una alternativa. Del mismo modo, si su proveedor de Internet tiene un servidor de tiempo, se recomienda utilizarlo en su lugar.

3. ¿Cómo podría un administrador de sistemas decidir unirse o contribuir de alguna manera al proyecto `pool.ntp.org`?

   Desde www.ntppool.org: Su servidor debe tener una dirección IP estática y una conexión permanente a Internet. La dirección IP estática no debe cambiar en absoluto o al menos menos una vez al año. Además, los requisitos de ancho de banda son modestos: 384 - 512 Kbit de ancho de banda. Los servidores de estrato 3 o 4 son bienvenidos.



108.2 Registros del sistema

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 108.2

Importancia

4

Áreas de conocimiento clave

- Configuración básica de rsyslog.
- Conocer los subsistemas, prioridades y acciones estándar.
- Hacer consultas al diario de systemd.
- Filtrar los datos del diario de systemd por criterios tales como la fecha, el servicio o la prioridad.
- Configurar el almacenamiento persistente del diario de systemd así como su tamaño.
- Borrar los datos antiguos del diario de systemd.
- Recuperar datos del diario de systemd a partir de un sistema de rescate o de una copia del sistema de archivos.
- Entender la interacción de rsyslog con systemd-journald.
- Configuración de logrotate.
- Conocer syslog y syslog-ng.

Lista parcial de archivos, términos y utilidades

- `/etc/rsyslog.conf`
- `/var/log/`

- `logger`
- `logrotate`
- `/etc/logrotate.conf`
- `/etc/logrotate.d/`
- `journalctl`
- `systemd-cat`
- `/etc/systemd/journald.conf`
- `/var/log/journal/`



108.2 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	108 Servicios esenciales del sistema
Objetivo:	108.2 Registros del sistema
Lección:	1 de 2

Introducción

Los registros pueden ser el mejor amigo de un administrador de sistemas. Son archivos (normalmente de texto) en los que se registran cronológicamente todos los eventos del sistema y de la red desde el momento en que se inicia. Así, la gama de información que se puede encontrar en los registros incluye prácticamente todos los aspectos del sistema: intentos fallidos de autenticación, errores de programas y servicios, hosts bloqueados por el cortafuegos, etc. Como puedes imaginar, los registros facilitan mucho la vida de los administradores de sistemas a la hora de solucionar problemas, comprobar recursos, detectar comportamientos anómalos de los programas, etc. En esta lección discutiremos una de las facilidades de registro más comunes que se encuentran actualmente en las distribuciones de GNU/Linux: `rsyslog`. Estudiaremos los diferentes tipos de logs que existen, dónde se almacenan, qué información incluyen y cómo se puede obtener y filtrar esa información. También discutiremos cómo se pueden mantener los logs en servidores centralizados a través de redes IP, la rotación de logs y el ring buffer del kernel.

Registro del sistema

En el momento en que el kernel y los diferentes procesos de tu sistema comienzan a ejecutarse y a comunicarse entre sí, se genera mucha información en forma de mensajes (que en su mayoría) se

envían a los registros.

Sin los registros, la búsqueda de un evento ocurrido en un servidor supondría un dolor de cabeza para los administradores del sistema, de ahí la importancia de contar con una forma estandarizada y centralizada de realizar un seguimiento de cualquier suceso del sistema. Los registros son determinantes y reveladores cuando se trata de la resolución de problemas y la seguridad, y son fuentes de datos fiables para entender las estadísticas del sistema y hacer predicciones de tendencias.

Dejando de lado `systemd-journald` (del que hablaremos en la próxima lección), el registro ha sido tradicionalmente manejado por tres servicios principales dedicados: `syslog`, `syslog-ng` (syslog new generation) y `rsyslog` ("`the rocket-fast system for log processing"). El `rsyslog` aportó importantes mejoras (como el soporte de RELP) y se ha convertido en la opción más popular hoy en día. Cada uno recoge mensajes de otros servicios y programas, y los almacena en archivos de registro, normalmente en `/var/log`. Sin embargo, algunos servicios se encargan de sus propios registros (por ejemplo, el servidor web Apache HTTPD o el sistema de impresión CUPS). Asimismo, el kernel de Linux utiliza el ring buffer para almacenar sus mensajes de registro.

NOTE `RELP` significa *Reliable Event Logging Protocol* y amplía la funcionalidad del protocolo syslog para proporcionar una entrega fiable de los mensajes.

Dado que `rsyslog` se ha convertido en la instalación de registro estándar *oficial* en todas las principales distros, nos centraremos en ella para la presente lección. `rsyslog` utiliza un modelo cliente-servidor. El cliente y el servidor pueden estar en el mismo host o en diferentes máquinas. Los mensajes son enviados y recibidos en un formato particular y pueden ser guardados en servidores centralizados de `rsyslog` a través de la red. El demonio de rsyslog — `rsyslogd` — trabaja junto con `klogd` (que gestiona los mensajes del kernel). En las próximas secciones se hablará de `rsyslog` y su infraestructura de registro.

NOTE Un demonio es un servicio que se ejecuta en segundo plano. Tenga en cuenta la "d" final en los nombres de los demonios: "klogd" o "rsyslogd".

Tipos de registros

Como los registros son datos *variables*, normalmente se encuentran en `/var/log`. A grandes rasgos, se pueden clasificar en *registros de sistema* y *registros de servicio o programa.*

Veamos algunos registros del sistema y la información que guardan: `/var/log/auth.log`:: Actividades relacionadas con los procesos de autenticación: usuarios registrados, información `sudo`, trabajos cron, intentos fallidos de inicio de sesión, etc.

`/var/log/syslog`

Es un archivo centralizado para prácticamente todos los registros capturados por `rsyslogd`. Debido a que incluye tanta información, los registros se distribuyen a través de otros archivos de acuerdo con la configuración suministrada en `/etc/rsyslog.conf`.

`/var/log/debug`

Información de depuración de programas.

`/var/log/kern.log`

Mensajes de kernel.

`/var/log/messages`

Mensajes informativos que no están relacionados con el kernel sino con otros servicios. También es el destino por defecto del registro del cliente remoto en una implementación de servidor de registro centralizado.

`/var/log/daemon.log`

Información relacionada con los demonios o servicios que se ejecutan en segundo plano.

`/var/log/mail.log`

Información relacionada con el servidor de correo electrónico, por ejemplo, postfix.

`/var/log/Xorg.0.log`

Información relacionada con la tarjeta gráfica.

`/var/run/utmp` and `/var/log/wtmp`

Registros de acceso exitosos.

`/var/log/btmp`

Intentos fallidos de inicio de sesión, por ejemplo, ataque de fuerza bruta a través de ssh.

`/var/log/faillog`

Intentos de autenticación fallidos.

`/var/log/lastlog`

Fecha y hora de los últimos inicios de sesión de los usuarios.

Veamos ahora algunos ejemplos de registros de servicio:

`/var/log/cups/`

Directorio para los registros del *Sistema de impresión (Common Unix Printing System)*. Normalmente incluye los siguientes archivos de registro por defecto: `error_log`, `page_log` y `access_log`.

`/var/log/apache2/` or `/var/log/httpd`

Directorio para los registros del *Servidor Web Apache*. Normalmente incluye los siguientes archivos de registro por defecto: `access.log`, `error_log`, y `other_vhosts_access.log`.

`/var/log/mysql`

Directorio para los registros del Sistema de Gestión de Bases de Datos Relacionales *MySQL*. Suele incluir los siguientes archivos de registro por defecto: `error_log`, `mysql.log` y `mysql-slow.log`.

`/var/log/samba/`

Directorio para los registros del protocolo *Session Message Block* (SMB). Suele incluir los siguientes archivos de registro por defecto: `log.`, `log.nmbd` and `log.smbd`.

NOTE El nombre exacto y el contenido de los archivos de registro pueden variar según la distribución de Linux. También hay registros particulares de distribuciones específicas como `/var/log/dpkg.log` (que contiene información relacionada con los paquetes `dpkg`) en Debian GNU/Linux y sus derivados.

Leyendo registros

Para leer los archivos de registro, primero asegúrese de que es el usuario root o de que tiene permisos de lectura sobre el archivo. Puede utilizar una variedad de utilidades como:

`less` o `more`

Permiten ver y desplazarse por una página a la vez:

```
root@debian:~# less /var/log/auth.log
Sep 12 18:47:56 debian sshd[441]: Received SIGHUP; restarting.
Sep 12 18:47:56 debian sshd[441]: Server listening on 0.0.0.0 port 22.
Sep 12 18:47:56 debian sshd[441]: Server listening on :: port 22.
Sep 12 18:47:56 debian sshd[441]: Received SIGHUP; restarting.
Sep 12 18:47:56 debian sshd[441]: Server listening on 0.0.0.0 port 22.
Sep 12 18:47:56 debian sshd[441]: Server listening on :: port 22.
Sep 12 18:49:46 debian sshd[905]: Accepted password for carol from 192.168.1.65
port 44296 ssh2
Sep 12 18:49:46 debian sshd[905]: pam_unix(sshd:session): session opened for
user carol by (uid=0)
Sep 12 18:49:46 debian systemd-logind[331]: New session 2 of user carol.
Sep 12 18:49:46 debian systemd: pam_unix(systemd-user:session): session opened
for user carol by (uid=0)
(...)
```

`zless` or `zmore`

Lo mismo que `less` y `more`, pero utilizado para los registros que se comprimen con `gzip` (una función común de `logrotate`):

```
root@debian:~# zless /var/log/auth.log.3.gz
Aug 19 20:05:57 debian sudo:    carol : TTY=pts/0 ; PWD=/home/carol ; USER=root
; COMMAND=/sbin/shutdown -h now
Aug 19 20:05:57 debian sudo: pam_unix(sudo:session): session opened for user
root by carol(uid=0)
Aug 19 20:05:57 debian lightdm: pam_unix(lightdm-greeter:session): session
closed for user lightdm
Aug 19 23:50:49 debian systemd-logind[333]: Watching system buttons on
/dev/input/event2 (Power Button)
Aug 19 23:50:49 debian systemd-logind[333]: Watching system buttons on
/dev/input/event3 (Sleep Button)
Aug 19 23:50:49 debian systemd-logind[333]: Watching system buttons on
/dev/input/event4 (Video Bus)
Aug 19 23:50:49 debian systemd-logind[333]: New seat seat0.
Aug 19 23:50:49 debian sshd[409]: Server listening on 0.0.0.0 port 22.
(...)
```

`tail`

Muestra las últimas líneas de un archivo (el valor por defecto es de 10 líneas). El poder de tail reside en el parámetro `-f`, que muestra dinámicamente las nuevas líneas a medida que se añaden:

```
root@suse-server:~# tail -f /var/log/messages
2019-09-14T13:57:28.962780+02:00 suse-server sudo: pam_unix(sudo:session):
session closed for user root
2019-09-14T13:57:38.038298+02:00 suse-server sudo:    carol : TTY=pts/0 ;
PWD=/home/carol ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/messages
2019-09-14T13:57:38.039927+02:00 suse-server sudo: pam_unix(sudo:session):
session opened for user root by carol(uid=0)
2019-09-14T14:07:22+02:00 debian carol: appending new message from client to
remote server...
```

`head`

Ver las primeras líneas de un archivo (por defecto son 10 líneas):

```
root@suse-server:~# head -5 /var/log/mail
2019-06-29T11:47:59.219806+02:00 suse-server postfix/postfix-script[1732]: the
Postfix mail system is not running
2019-06-29T11:48:01.355361+02:00 suse-server postfix/postfix-script[1925]:
starting the Postfix mail system
2019-06-29T11:48:01.391128+02:00 suse-server postfix/master[1930]: daemon
started -- version 3.3.1, configuration /etc/postfix
2019-06-29T11:55:39.247462+02:00 suse-server postfix/postfix-script[3364]:
stopping the Postfix mail system
2019-06-29T11:55:39.249375+02:00 suse-server postfix/master[1930]: terminating
on signal 15
```

`grep`

Utilidad de filtrado que permite buscar cadenas específicas:

```
root@debian:~# grep "dhclient" /var/log/syslog
Sep 13 11:58:48 debian dhclient[448]: DHCPREQUEST of 192.168.1.4 on enp0s3 to
192.168.1.1 port 67
Sep 13 11:58:49 debian dhclient[448]: DHCPACK of 192.168.1.4 from 192.168.1.1
Sep 13 11:58:49 debian dhclient[448]: bound to 192.168.1.4 -- renewal in 1368
seconds.
(...)
```

Como habrá notado, la salida se imprime en el siguiente formato:

- Marca de tiempo
- Nombre del host desde el que se originó el mensaje

- Nombre del programa/servicio que ha generado el mensaje
- El PID del programa que generó el mensaje
- Descripción de la acción realizada

Hay algunos ejemplos en los que los registros no son de texto, sino archivos binarios, y por consiguiente, hay que utilizar comandos especiales para analizarlos:

`/var/log/wtmp`

Use `who` (or `w`):

```
root@debian:~# who
root     pts/0        2020-09-14 13:05 (192.168.1.75)
root     pts/1        2020-09-14 13:43 (192.168.1.75)
```

`/var/log/btmp`

Use `utmpdump` o `last -f`:

```
root@debian:~# utmpdump /var/log/btmp
Utmp dump of /var/log/btmp
[6] [01287] [    ] [dave     ] [ssh:notty   ] [192.168.1.75        ]
[192.168.1.75   ] [2019-09-07T19:33:32,000000+0000]
```

`/var/log/faillog`

Use `faillog`:

```
root@debian:~# faillog -a | less
Login       Failures Maximum Latest                   On

root            0        0   01/01/70 01:00:00 +0100
daemon          0        0   01/01/70 01:00:00 +0100
bin             0        0   01/01/70 01:00:00 +0100
sys             0        0   01/01/70 01:00:00 +0100
sync            0        0   01/01/70 01:00:00 +0100
games           0        0   01/01/70 01:00:00 +0100
man             0        0   01/01/70 01:00:00 +0100
lp              0        0   01/01/70 01:00:00 +0100
mail            0        0   01/01/70 01:00:00 +0100
(...)
```

/var/log/lastlog

Use `lastlog`:

```
root@debian:~# lastlog | less
Username         Port     From             Latest
root                                       Never logged in
daemon                                     Never logged in
bin                                        Never logged in
sys                                        Never logged in
(...)
sync                                       Never logged in
avahi                                      Never logged in
colord                                     Never logged in
saned                                      Never logged in
hplip                                      Never logged in
carol            pts/1    192.168.1.75     Sat Sep 14 13:43:06 +0200 2019
dave             pts/3    192.168.1.75     Mon Sep  2 14:22:08 +0200 2019
```

NOTE También hay herramientas gráficas para leer los archivos de registro, por ejemplo: `gnome-logs` y `KSystemLog`.

¿Cómo se convierten los mensajes en registros?

El siguiente proceso ilustra cómo se escribe un mensaje en un archivo de registro:

1. Las aplicaciones, los servicios y el kernel escriben mensajes en archivos especiales (sockets y buffers de memoria), por ejemplo `/dev/log` o `/dev/kmsg`.
2. `rsyslogd` obtiene la información de los sockets o buffers de memoria.
3. Dependiendo de las reglas encontradas en `/etc/rsyslog.conf` y/o de los archivos en `/etc/ryslog.d/`, `rsyslogd` mueve la información al archivo de registro correspondiente (típicamente encontrado en `/var/log`).

NOTE Un socket es un archivo especial utilizado para transferir información entre diferentes procesos. Para listar todos los sockets de su sistema, puede utilizar el comando `systemctl list-sockets --all`.

Facilidades, prioridades y acciones

El archivo de configuración de `rsyslog` es `/etc/rsylog.conf` (en algunas distribuciones también

puede encontrar archivos de configuración en `/etc/rsyslog.d/`). Normalmente se divide en tres secciones: `MODULES`, `GLOBAL DIRECTIVES` y `RULES`. Vamos a echarle un vistazo explorando el fichero `rsyslog.conf` en nuestro host Debian GNU/Linux 10 (buster) — para hacer esto puede usar `sudo less /etc/rsyslog.conf`.

`MODULES` incluye el soporte de módulos para el registro, la capacidad de mensajes y la recepción de registros UDP/TCP:

```
#################
#### MODULES ####
#################

module(load="imuxsock") # proporciona soporte para el registro del sistema local
module(load="imklog")   # proporciona soporte de registro del kernel
#module(load="immark")  # proporciona la capacidad de mensajes --MARK--.

# proporciona la recepción de syslogs UDP
#module(load="imudp")
#input(type="imudp" port="514")

# proporciona la recepción de syslogs TCP
#module(load="imtcp")
#input(type="imtcp" port="514")
```

`GLOBAL DIRECTIVES` permiten configurar una serie de cosas como los registros y los permisos del directorio de registros:

```
###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# UtilicE el formato tradicional de marca de tiempo.
#  Para habilitar las marcas de tiempo de alta precisión, comenta la siguiente
línea.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

#
# Establezca los permisos por defecto para todos los archivos de registro.
#
$FileOwner root
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022

#
# Dónde colocar los archivos de spool y de estado
#
$WorkDirectory /var/spool/rsyslog

#
# Incluir todos los archivos de configuración en /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf
```

`RULES` es donde entran las *facilidades*, las *prioridades* y las *acciones*. Las configuraciones de esta sección le dicen al demonio de registro que filtre los mensajes de acuerdo con ciertas reglas y los registre o envíe cuando sea necesario. Para comprender estas reglas, primero debemos explicar los conceptos de facilidades y prioridades de `rsyslog`. A cada mensaje de registro se le da un número de *facilidad* y una palabra clave que están asociados con el subsistema interno de Linux que produce el mensaje:

Número	**Palabra clave**	**Descripción**
`0`	`kern`	Mensajes del kernel de Linux
`1`	`user`	Mensajes a nivel de usuario
`2`	`mail`	Mensajes del sistema de correo

Número	Palabra clave	Descripción
`3`	`daemon`	Demonios del sistema
`4`	`auth`, `authpriv`	Mensajes de seguridad/autorización
`5`	`syslog`	Mensajes de syslogd
`6`	`lpr`	Subsistema de impresión de línea
`7`	`news`	Mensajes del subsitema de red
`8`	`uucp`	Subsistema UUCP (Unix-to-Unix Copy Protocol)
`9`	`cron`	Demonio del reloj
`10`	`auth`, `authpriv`	Mensajes de seguridad/autorización
`11`	`ftp`	Demonio del FTP
`12`	`ntp`	Demonio del NTP
`13`	`security`	Registros de auditoría
`14`	`console`	Registros de alertas
`15`	`cron`	Demonio del cron
`16 – 23`	De `local0` al `local7`	Local utiliza 0 - 7

Además, a cada mensaje se le asigna un nivel de *prioridad*:

Código	Severidad	Palabra clave	Descripción
`0`	Emergency	`emerg`, `panic`	El sistema es inutilizable
`1`	Alert	`alert`	Hay que actuar inmediatamente
`2`	Critical	`crit`	Condiciones críticas
`3`	Error	`err`, `error`	Condiciones de error
`4`	Warning	`warn`, `warning`	Condiciones de advertencia

Código	**Severidad**	**Palabra clave**	**Descripción**
`5`	Notice	`notice`	Condición normal pero significativa
`6`	Informational	`info`	Mensajes informativos
`7`	Debug	`debug`	Mensajes de nivel de depuración

Aquí hay un extracto de `rsyslog.conf` de nuestro sistema Debian GNU/Linux 10 (buster) que incluye algunas reglas de ejemplo:

```
###############
#### RULES ####
###############

# Primero algunos archivos de registro estándar.  Registro por instalación.
#
auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log

#
# Registro para el sistema de correo.  Dividirlo para que
# sea fácil escribir scripts para analizar estos archivos.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err

#
# Algunos archivos de registro "catch-all"
#
*.=debug;\
        auth,authpriv.none;\
    news.none;mail.none     -/var/log/debug
*.=info;*.=notice;*.=warn;\
    auth,authpriv.none;\
    cron,daemon.none;\
    mail,news.none          -/var/log/messages
```

El formato de la regla es el siguiente:`<facilidad>.<prioridad>` <acción>

Los selectores <facilidad>.<prioridad> filtran los mensajes que deben coincidir. Los niveles de prioridad son jerárquicamente inclusivos, lo que significa que rsyslog coincidirá con los mensajes de la prioridad especificada y superiores. El selector <acción> muestra la acción a realizar (dónde enviar el mensaje de registro). Aquí hay algunos ejemplos para mayor claridad:

```
auth,authpriv.*                 /var/log/auth.log
```

Independientemente de su prioridad (`*`), todos los mensajes de `auth` or `authpriv` serán enviados a `/var/log/auth.log`.

```
*.*;auth,authpriv.none          -/var/log/syslog
```

Todos los mensajes — independientemente de su prioridad (`*`) — de todas las facilidades (`*`) — descartando los de `auth` o `authpriv` (de ahí el sufijo `.none`) — se escribirán en `/var/log/syslog` (el signo menos (`-`) antes de la ruta evita excesivas escrituras en disco). Tenga en cuenta el punto y coma (`;`) para dividir el selector y la coma (`,`) para concatenar dos instalaciones en la misma regla (`auth,authpriv`).

```
mail.err                        /var/log/mail.err
```

Los mensajes de la instalación de `mail` con un nivel de prioridad de `error` o superior (`crítico`, `alerta` o `emergencia`) se enviarán a `/var/log/mail.err`.

```
*.=debug;\
        auth,authpriv.none;\
    news.none;mail.none     -/var/log/debug
```

Los mensajes de todas las instalaciones con la prioridad `debug` y ninguna otra (`=`) se escribirán en `/var/log/debug` — excluyendo cualquier mensaje procedente de las instalaciones `auth`, `authpriv`, `news` y `mail` (nótese la sintaxis: `;\`).

Entradas manuales en el registro del sistema: `logger`

El comando `logger` es muy útil para los scripts de la shell o para propósitos de prueba. El comando `logger` agregará cualquier mensaje que reciba a `/var/log/syslog` (o a `/var/log/messages` cuando se registre en un servidor de registro central remoto, como se verá más adelante en esta lección):

```
carol@debian:~$ logger this comment goes into "/var/log/syslog"
```

Para imprimir la última línea en `/var/log/syslog`, utilice el comando `tail` con la opción `-1`:

```
root@debian:~# tail -1 /var/log/syslog
Sep 17 17:55:33 debian carol: this comment goes into /var/log/syslog
```

`rsyslog` como servidor central de registros

Para explicar este tema vamos a añadir un nuevo host a nuestra configuración. La disposición es la siguiente:

Role	Hostname	S.O	IP Address
Central Log Server	`suse-server`	openSUSE Leap 15.1	192.168.1.6
Client	`debian`	Debian GNU/Linux 10 (buster)	192.168.1.4

Empecemos por configurar el servidor. En primer lugar, nos aseguramos de que `rsyslog` este en funcionamiento:

```
root@suse-server:~# systemctl status rsyslog
 rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; vendor preset:
enabled)
   Active: active (running) since Thu 2019-09-17 18:45:58 CEST; 7min ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
 Main PID: 832 (rsyslogd)
    Tasks: 5 (limit: 4915)
   CGroup: /system.slice/rsyslog.service
           ␣␣832 /usr/sbin/rsyslogd -n -iNONE
```

openSUSE incluye un archivo de configuración dedicado al registro remoto: `/etc/rsyslog.d/remote.conf`. Vamos a activar la recepción de mensajes de los clientes (hosts remotos) a través de TCP. Debemos descomentar las líneas que cargan el módulo e inician el servidor TCP en el puerto 514:

```
# ######### Recepción de mensajes de hosts remotos ##########
# TCP Syslog Server:
# provides TCP syslog reception and GSS-API (if compiled to support it)
$ModLoad imtcp.so  # load module
##$UDPServerAddress 10.10.0.1  # force to listen on this IP only
$InputTCPServerRun 514  # Starts a TCP server on selected port

# UDP Syslog Server:
#$ModLoad imudp.so  # provides UDP syslog reception
##$UDPServerAddress 10.10.0.1  # force to listen on this IP only
#$UDPServerRun 514  # start a UDP syslog server at standard port 514
```

Una vez hecho esto, debemos reiniciar el servicio rsyslog y comprobar que el servidor está escuchando en el puerto 514:

```
root@suse-server:~# systemctl restart rsyslog
root@suse-server:~# netstat -nltp | grep 514
[sudo] password for root:
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN
2263/rsyslogd
tcp6       0      0 :::514                  :::*                    LISTEN
2263/rsyslogd
```

A continuación, debemos abrir los puertos en el firewall y recargar la configuración:

```
root@suse-server:~# firewall-cmd --permanent --add-port 514/tcp
success
root@suse-server:~# firewall-cmd --reload
success
```

NOTE Con la llegada de openSUSE Leap 15.0, `firewalld` sustituyó por completo al clásico `SuSEFirewall2`.

Plantillas y condiciones de filtrado

Por defecto, los registros del cliente se escribirán en el archivo `/var/log/messages` del servidor, junto con los del propio servidor. Sin embargo, crearemos una *plantilla* y una *condición de filtro* para que los registros de nuestro cliente se almacenen en directorios propios. Para ello, añadiremos lo siguiente a `/etc/rsyslog.conf` (o `/etc/rsyslog.d/remote.conf`):

```
$template RemoteLogs,"/var/log/remotehosts/%HOSTNAME%/%$NOW%.%syslogseverity-
text%.log"
if $FROMHOST-IP=='192.168.1.4' then ?RemoteLogs
& stop
```

Template

La plantilla corresponde a la primera línea y permite especificar un formato para los nombres de registro mediante la generación dinámica de nombres de archivo. Una plantilla se compone de:

- Directiva de plantillas (`$template`)
- Nombre de la plantilla (`RemoteLogs`)
- Texto de la plantilla (`"/var/log/remotehosts/%HOSTNAME%/%$NOW%.%syslogseverity-text%.log"`)
- Opciones (optional)

Nuestra plantilla se llama `RemoteLogs` y su texto consiste en una ruta en `/var/log`. Todos los registros de nuestro host remoto irán al directorio `remotehosts`, donde se creará un subdirectorio basado en el nombre del host de la máquina (`%HOSTNAME%`). Cada nombre de archivo en este directorio consistirá en la fecha (`%$NOW%`), la gravedad (también conocida como prioridad) del mensaje en formato de texto (`%syslogseverity-text%`) y el sufijo `.log`. Las palabras entre los signos de porcentaje son *propiedades* y permiten acceder al contenido del mensaje de registro (fecha, prioridad, etc.). Un mensaje `syslog` tiene una serie de propiedades bien definidas que pueden utilizarse en las plantillas. A estas propiedades se accede -y se pueden modificar- mediante el llamado *reemplazador de propiedades* que implica escribirlas entre signos de porcentaje.

Condición del filtro

Las dos líneas restantes corresponden a la condición del filtro y su acción asociada:

- Filtro basado en expresiones (`if $FROMHOST-IP=='192.168.1.4'`)
- Acción (`then ?RemoteLogs`, `& stop`)

La primera línea comprueba la dirección IP del host remoto que envía el registro y — si es igual a la de nuestro cliente Debian — aplica la plantilla `RemoteLogs`. La última línea (`& stop`) garantiza que los mensajes no se envían simultáneamente a `/var/log/messages` (sino sólo a los ficheros del directorio `/var/log/remotehosts`).

NOTE Para saber más sobre plantillas, propiedades y reglas, puedes consultar la página del manual de `rsyslog.conf`.

Con la configuración actualizada, reiniciaremos `rsyslog` de nuevo y confirmaremos que aún no existe el directorio `remotehosts` en `/var/log`:

```
root@suse-server:~# systemctl restart rsyslog
root@suse-server:~# ls /var/log/
acpid              chrony     localmessages   pbl.log          Xorg.0.log
alternatives.log   cups       mail            pk_backend_zypp  Xorg.0.log.old
apparmor           firebird   mail.err        samba            YaST2
audit              firewall   mail.info       snapper.log      zypp
boot.log           firewalld  mail.warn       tallylog         zypper.log
boot.msg           krb5       messages        tuned
boot.omsg          lastlog    mysql           warn
btmp               lightdm    NetworkManager  wtmp
```

El servidor ya está configurado. A continuación, configuraremos el cliente.

De nuevo, debemos asegurarnos de que `rsyslog` está instalado y funcionando:

```
root@debian:~# sudo systemctl status rsyslog
 rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset:
   Active: active (running) since Thu 2019-09-17 18:47:54 CEST; 7min ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
 Main PID: 351 (rsyslogd)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/rsyslog.service
           351 /usr/sbin/rsyslogd -n
```

En nuestro entorno de ejemplo hemos implementado la resolución de nombres en el cliente añadiendo la línea `192.168.1.6 suse-server` a `/etc/hosts`. Así podemos referirnos al servidor tanto por su nombre (`suse-server`) como por su dirección IP (`192.168.1.6`).

Nuestro cliente Debian no viene con un fichero `remote.conf` en `/etc/rsyslog.d/`, así que aplicaremos nuestras configuraciones en `/etc/rsyslog.conf`. Escribiremos la siguiente línea al final del fichero:

```
*.* @@suse-server:514
```

Por último, reiniciamos `rsyslog`.

```
root@debian:~# systemctl restart rsyslog
```

Ahora, volvamos a nuestra máquina `suse-server` y comprobemos la existencia de `remotehosts` en `/var/log`:

```
root@suse-server:~# ls /var/log/remotehosts/debian/
2019-09-17.info.log  2019-09-17.notice.log
```

Ya tenemos dos registros dentro de `/var/log/remotehosts` como se describe en nuestra plantilla. Para completar esta sección, ejecutamos `tail -f 2019-09-17.notice.log` en `suse-server` mientras enviamos un registro *manualmente* desde nuestro cliente Debian y confirmamos que los mensajes se añaden al archivo de registro como se esperaba (la opción `-t` proporciona una etiqueta para nuestro mensaje):

```
root@suse-server:~# tail -f /var/log/remotehosts/debian/2019-09-17.notice.log
2019-09-17T20:57:42+02:00 debian dbus[323]: [system] Successfully activated service
'org.freedesktop.nm_dispatcher'
2019-09-17T21:01:41+02:00 debian anacron[1766]: Anacron 2.3 started on 2019-09-17
2019-09-17T21:01:41+02:00 debian anacron[1766]: Normal exit (0 jobs run)
```

```
carol@debian:~$ logger -t DEBIAN-CLIENT Hi from 192.168.1.4
```

```
root@suse-server:~# tail -f /var/log/remotehosts/debian/2019-09-17.notice.log
2019-09-17T20:57:42+02:00 debian dbus[323]: [system] Successfully activated service
'org.freedesktop.nm_dispatcher'
2019-09-17T21:01:41+02:00 debian anacron[1766]: Anacron 2.3 started on 2019-09-17
2019-09-17T21:01:41+02:00 debian anacron[1766]: Normal exit (0 jobs run)
2019-09-17T21:04:21+02:00 debian DEBIAN-CLIENT: Hi from 192.168.1.4
```

Mecanismo de rotación de registros

Los registros se rotan con regularidad, lo que sirve para dos propósitos principales: * Evitar que los archivos de registro más antiguos utilicen más espacio de disco del necesario.

- Mantenga los registros con una longitud manejable para facilitar su consulta.

La utilidad encargada de la rotación (o ciclado) de los registros es `logrotate` y su trabajo incluye acciones como mover los archivos de registro a un nuevo nombre, archivarlos y/o comprimirlos, a

veces enviarlos por correo electrónico al administrador del sistema y eventualmente borrarlos a medida que envejecen. Hay varias convenciones para nombrar estos archivos de registro rotados (añadiendo un sufijo con la fecha al nombre del archivo, por ejemplo); sin embargo, la práctica común es simplemente añadir un sufijo con un número entero:

```
root@debian:~# ls /var/log/messages*
/var/log/messages  /var/log/messages.1  /var/log/messages.2.gz
/var/log/messages.3.gz  /var/log/messages.4.gz
```

Expliquemos ahora lo que ocurrirá en la próxima rotación del registro:

1. `messages.4.gz` se eliminará.
2. El contenido de `messages.3.gz` se trasladará a `messages.4.gz`.
3. El contenido de `messages.2.gz` se trasladará a `messages.3.gz`.
4. El contenido de `messages.1` se trasladará a `messages.2.gz`.
5. El contenido de `messages` se trasladará a `messages.1` y `messages` estará vacío y listo para registrar nuevas entradas de registro.

Observe que según las directivas `logrotate` (que verá en breve), los tres archivos de registro más antiguos están comprimidos, mientras que los dos más recientes no lo están. Además, conservaremos los registros de las últimas 4-5 semanas. Para leer los mensajes de hace una semana, consultaremos `messages.1` (y así sucesivamente).

`logrotate` se ejecuta como un proceso automatizado o trabajo cron diariamente a través del script `/etc/cron.daily/logrotate` y lee el archivo de configuración `/etc/logrotate.conf`. Este archivo incluye algunas opciones globales y está bien comentado con cada opción introducida por una breve explicación de su propósito:

```
carol@debian:~$ sudo less /etc/logrotate.conf
# see "man logrotate" for details
# rotar los archivos de registro semanalmente
weekly

# mantener 4 semanas
rotate 4

# crear nuevos archivos de registro (vacíos) después de rotar los antiguos
create

# Descomente esto si quiere que sus archivos de registro se compriman
#compress

# los paquetes dejan la información de la rotación del registro en este directorio
include /etc/logrotate.d

(...)
```

Como puedes ver, también se incluyen los archivos de configuración en `/etc/logrotate.d` para paquetes específicos. Estos archivos contienen — en su mayoría — definiciones locales y especifican los archivos de registro a rotar (recuerde, las definiciones locales tienen prioridad sobre las globales, y las definiciones posteriores anulan las anteriores). Lo que sigue es un extracto de una definición en `/etc/logrotate.d/rsyslog`:

```
/var/log/messages
{
        rotate 4
        weekly
        missingok
        notifempty
        compress
        delaycompress
        sharedscripts
        postrotate
                invoke-rc.d rsyslog rotate > /dev/null
        endscript
}
```

Como puede ver, cada directiva está separada de su valor por espacios en blanco y/o un signo de igualdad opcional (`=`). Sin embargo, las líneas entre `postrotate` y `endscript` deben aparecer en líneas solas. La explicación es la siguiente:

`rotate 4`

Conserva los registros de 4 semanas.

`weekly`

Rota los archivos de registro semanalmente.

`missingok`

No emita un mensaje de error si falta el archivo de registro; simplemente pase al siguiente.

`notifempty`

No gira el registro si está vacío.

`compress`

Comprime los archivos de registro con `gzip` (por defecto).

`delaycompress`

Pospone la compresión del archivo de registro anterior al siguiente ciclo de rotación (sólo es efectivo cuando se utiliza en combinación con compress). Es útil cuando no se puede indicar a un programa que cierre su archivo de registro y, por tanto, podría seguir escribiendo en el archivo anterior durante algún tiempo.

`sharedscripts`

Relacionado con los scripts *prerotate* y *postrotate*. Para evitar que un script se ejecute varias veces, ejecute los scripts sólo una vez, independientemente de cuántos archivos de registro coincidan con un patrón determinado (por ejemplo, `/var/log/mail/*`). Además, si los scripts salen con errores, las acciones restantes no se ejecutarán para ningún registro.

`postrotate`

Indica el inicio de un script *postrotate*.

`invoke-rc.d rsyslog rotate > /dev/null`

Utiliza `/bin/sh` para correr `invoke-rc.d rsyslog rotate > /dev/null` después de rotar los registros.

`endscript`

Indica el final del script *postrotate* .

NOTE Para una lista completa de directivas y explicaciones, consulte la página del manual de `logrotate.conf`.

Kernel Ring Buffer

Dado que el kernel genera varios mensajes antes de que `rsyslogd` esté disponible en el arranque, se hace necesario un mecanismo para registrar esos mensajes. Aquí es donde entra en juego el *kernel ring buffer*. Es una estructura de datos de tamaño fijo y, por lo tanto, a medida que crece con nuevos mensajes, los más antiguos desaparecen.

El comando `dmesg` imprime el kernel ring buffer. Debido al tamaño del buffer, este comando se utiliza normalmente en combinación con la utilidad de filtrado de texto `grep`. Por ejemplo, para buscar mensajes relacionados con los dispositivos del Bus Serie Universal:

```
root@debian:~# dmesg | grep "usb"
[    1.241182] usbcore: registered new interface driver usbfs
[    1.241188] usbcore: registered new interface driver hub
[    1.250968] usbcore: registered new device driver usb
[    1.339754] usb usb1: New USB device found, idVendor=1d6b, idProduct=0001,
bcdDevice= 4.19
[    1.339756] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
(...)
```

Ejercicios guiados

1. Qué utilidades/comandos utilizarías en los siguientes escenarios:

Finalidad y archivo de registro	**Utilidad**
Leer `/var/log/syslog.7.gz`	
Leer `/var/log/syslog`	
Filtrar la palabra `renewal` en `/var/log/syslog`	
Leer `/var/log/faillog`	
Leer `/var/log/syslog` dinamicamente	

2. Reorganice las siguientes entradas de registro de manera que representen un mensaje de registro válido con la estructura adecuada:

   - `debian-server`
   - `sshd`
   - `[515]:`
   - `Sep 13 21:47:56`
   - `Server listening on 0.0.0.0 port 22`

   El orden correcto es:

3. Qué reglas añadirías a `/etc/rsyslog.conf` para cumplir con cada una de las siguientes:

   - Enviar todos los mensajes de la instalación `mail` y una prioridad/gravedad de `crit` (y superior) a `/var/log/mail.crit`:

   - Envía todos los mensajes de la instalación `mail` con prioridades de `alerta` y `emergencia` a `/var/log/mail.urgent`:

   - Excepto los procedentes de las instalaciones `cron` y `ntp`, envía todos los mensajes -independientemente de su facilidades y prioridad - a `/var/log/allmessages`:

- Con todos los ajustes requeridos correctamente configurados primero, envíe todos los mensajes de la instalación `mail` a un host remoto cuya dirección IP es `192.168.1.88` usando TCP y especificando el puerto por defecto:

```

```

- Independientemente de su facilidad, envía todos los mensajes con la prioridad `warning` (*sólo con la prioridad* `warning`*`) a `/var/log/warnings* evitando la escritura excesiva en el disco:

```

```

4. Considere la siguiente sección de `/etc/logrotate.d/samba` y explique las diferentes opciones:

```
carol@debian:~$ sudo head -n 11 /etc/logrotate.d/samba
/var/log/samba/log.smbd {
        weekly
        missingok
        rotate 7
        postrotate
                [ ! -f /var/run/samba/smbd.pid ] || /etc/init.d/smbd reload >
/dev/null
        endscript
        compress
        delaycompress
        notifempty
}
```

Opción	Significado
`weekly`	
`missingok`	
`rotate 7`	
`postrotate`	
`endscript`	
`compress`	
`delaycompress`	
`notifempty`	

Ejercicios de exploración

1. En la sección “Plantillas y condiciones de filtrado” hemos utilizado *uno basado en expresiones* como condición de filtrado. Los *filtros basados en propiedades* son otro tipo de exclusivo de `rsyslogd`. Convierta nuestro *filtro basado en expresiones* en *uno basado en propiedades*:

Filtro basado en expresiones	**Filtro basado en propiedades**
`if $FROMHOST-IP=='192.168.1.4' then ?RemoteLogs`	

2. `omusrmsg` es un módulo integrado en `rsyslog` que facilita la notificación a los usuarios (envía mensajes de registro al terminal del usuario). Escribe una regla para enviar todos los mensajes de *emergencia* de todas las instalaciones tanto a `root` como al usuario regular `carol`.

Resumen

En esta lección aprendió:

- El registro es crucial para la administración del sistema.
- `rsyslogd` es la utilidad encargada de mantener los registros limpios y ordenados.
- Algunos servicios se encargan de sus propios registros.
- A grandes rasgos, los registros pueden clasificarse en registros de sistema y registros de servicio/programa.
- Hay un número de utilidades que son convenientes para la lectura de registros: `less`, `more`, `zless`, `zmore`, `grep`, `head` y `tail`.
- La mayoría de los archivos de registro son de texto plano; sin embargo, hay un pequeño número de archivos de registro binarios.
- En cuanto a los registros, `rsyslogd` recibe la información relevante de archivos especiales (sockets y buffers de memoria) antes de procesarla.
- Para clasificar los registros, `rsyslogd` utiliza reglas en `/etc/rsyslog.conf` o `/etc/rsyslog.d/*`.
- Cualquier usuario puede introducir sus propios mensajes en el registro del sistema manualmente con la utilidad `logger`.
- `rsyslog` permite mantener todos los registros de las redes IP en un servidor de registros centralizado.
- Las plantillas son útiles para formatear los nombres de los archivos de registro de forma dinámica.
- El objetivo de la rotación de registros es doble: evitar que los registros antiguos utilicen un espacio de disco excesivo y hacer que los registros de consulta sean manejables.

Respuesta a los ejercicios guiados

1. Qué utilidades/comandos utilizarías en los siguientes escenarios:

Finalidad y archivo de registro	Utilidad
Leer `/var/log/syslog.7.gz`	`zmore` or `zless`
Leer `/var/log/syslog`	`more` or `less`
Filtrar la palabra `renewal` en `/var/log/syslog`	`grep`
Leer `/var/log/faillog`	`faillog -a`
Leer `/var/log/syslog` dinamicamente	`tail -f`

2. Reorganice las siguientes entradas de registro de manera que representen un mensaje de registro válido con la estructura adecuada:
   - `debian-server`
   - `sshd`
   - `[515]:`
   - `Sep 13 21:47:56`
   - `Server listening on 0.0.0.0 port 22`

   El orden correcto es:

```
Sep 13 21:47:56 debian-server sshd[515]: Server listening on 0.0.0.0 port
22
```

3. Qué reglas añadirías a `/etc/rsyslog.conf` para cumplir con cada una de las siguientes:
   - Enviar todos los mensajes de la instalación `mail` y una prioridad/gravedad de `crit` (y superior) a `/var/log/mail.crit`:

```
mail.crit                 /var/log/mail.crit
```

   - Envía todos los mensajes de la instalación `mail` con prioridades de `alerta` y `emergencia` a `/var/log/mail.urgent`:

```
mail.alert                       /var/log/mail.urgent
```

- Excepto los procedentes de las instalaciones `cron` y `ntp`, envía todos los mensajes -independientemente de su facilidades y prioridad - a `/var/log/allmessages`:

```
*.*;cron.none;ntp.none                  /var/log/allmessages
```

- Con todos los ajustes requeridos correctamente configurados primero, envíe todos los mensajes de la instalación `mail` a un host remoto cuya dirección IP es `192.168.1.88` usando TCP y especificando el puerto por defecto:

```
mail.* @@192.168.1.88:514
```

- Independientemente de su facilidad, envía todos los mensajes con la prioridad `warning` (*sólo con la prioridad* `warning``) *a `/var/log/warnings* evitando la escritura excesiva en el disco:

```
*.=warning                         -/var/log/warnings
```

4. Considere la siguiente sección de `/etc/logrotate.d/samba` y explique las diferentes opciones:

```
carol@debian:~$ sudo head -n 11 /etc/logrotate.d/samba
/var/log/samba/log.smbd {
        weekly
        missingok
        rotate 7
        postrotate
                [ ! -f /var/run/samba/smbd.pid ] || /etc/init.d/smbd reload >
/dev/null
        endscript
        compress
        delaycompress
        notifempty
}
```

Opción	Significado
`weekly`	Rotar los archivos de registro semanalmente.
`missingok`	No emita un mensaje de error si falta el registro; simplemente continúe con el siguiente.

Opción	Significado
`rotate 7`	Mantenga 7 semanas de atrasos.
`postrotate`	Ejecute el script en la siguiente línea después de rotar los registros.
`endscript`	Indica el final de la secuencia de comandos *postrotate*.
`compress`	Comprime los registros con `gzip`.
`delaycompress`	En combinación con `comprimir`, pospone la compresión al siguiente ciclo de rotación.
`notifyempty`	No gire el registro si está vacío.

Respuestas a los ejercicios de exploración

1. En la sección “Plantillas y condiciones de filtrado” hemos utilizado *uno basado en expresiones* como condición de filtrado. Los *filtros basados en propiedades* son otro tipo de exclusivo de `rsyslogd`. Convierta nuestro *filtro basado en expresiones* en *uno basado en propiedades*:

Filtro basado en expresiones	Filtro basado en propiedades
`if $FROMHOST-IP=='192.168.1.4' then ?RemoteLogs`	`:fromhost-ip, isequal, "192.168.1.4" ?RemoteLogs`

2. `omusrmsg` es un módulo integrado en `rsyslog` que facilita la notificación a los usuarios (envía mensajes de registro al terminal del usuario). Escribe una regla para enviar todos los mensajes de *emergencia* de todas las instalaciones tanto a `root` como al usuario regular `carol`.

```
*.emerg                        :omusrmsg:root,carol
```



108.2 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	108 Servicios esenciales del sistema
Objetivo:	108.2 Registros del sistema
Lección:	2 de 2

Introducción

Con la adopción generalizada de `systemd` por parte de las principales distribuciones, el demonio del journal (`systemd-journald`) se ha convertido en el servicio de registro estándar. En esta lección discutiremos cómo opera y cómo puedes usarlo para hacer un gran número de cosas: consultarlo, filtrar su información por diferentes criterios, configurar su almacenamiento y tamaño, borrar datos viejos, recuperar sus datos desde un sistema de rescate o una copia del sistema de archivos y - por último pero no menos importante - entender su interacción con `rsyslogd`.

Fundamentos de `systemd`

Introducido por primera vez en Fedora, `systemd` ha sustituido progresivamente a SysV Init como gestor de sistemas y servicios *de hecho* en la mayoría de las principales distribuciones de Linux. Entre sus puntos fuertes están los siguientes:

- Facilidad de configuración: Archivos de unidad en comparación a los scripts SysV Init.
- Gestión versátil: Además de demonios y procesos, también gestiona dispositivos, sockets y puntos de montaje.
- Compatibilidad con SysV Init y Upstart.

- Carga paralela durante el arranque: los servicios se cargan en paralelo, en lugar de que Sysv Init los cargue secuencialmente.
- Cuenta con un servicio de registro llamado *journal* que presenta las siguientes ventajas:
  - Centraliza todos los registros en un solo lugar.
  - No requiere rotación de registros.
  - Los registros pueden ser deshabilitados, cargados en RAM o hechos persistentes.

Unidades y objetivos

`systemd` opera sobre *unidades (units)*. Una unidad es cualquier recurso que `systemd` puede gestionar (por ejemplo, red, bluetooth, etc.). Las unidades, a su vez, se rigen por *ficheros de unidades*. Estos son archivos de texto plano que se ubican en `/lib/systemd/system` e incluyen los ajustes de configuración — en forma de *secciones* y *directivas* — para un recurso particular a ser gestionado. Hay varios tipos de unidades: `service`, `mount`, `automount`, `swap`, `timer`, `device`, `socket`, `path`, `timer`, `snapshot`, `slice`, `scope` y `target`. Así cada nombre de archivo de unidad sigue el patrón `<nombre_de_recurso>.<tipo_de_unidad>` (por ejemplo, `reboot.service`).

Un *objetivo (target)* es un tipo especial de unidad que se asemeja a los clásicos runlevels de SysV Init. Esto se debe a que una unidad *target* reúne varios recursos para representar un estado particular del sistema (por ejemplo, `graphical.target` es similar a `runlevel 5`, etc.). Para comprobar el objetivo actual de su sistema, utilice el comando `systemctl get-default`:

```
carol@debian:~$ systemctl get-default
graphical.target
```

Por otro lado, los objetivos y los niveles de ejecución se diferencian en que los primeros se incluyen mutuamente, mientras que los segundos no. Así, un objetivo puede hacer que aparezcan otros objetivos, lo que no es posible con los niveles de ejecución.

NOTE Una explicación de cómo funcionan las unidades `systemd` está fuera del alcance de esta lección.

The System Journal: `systemd-journald`

`systemd-journald` es el servicio del sistema que se encarga de recibir información de registro de diversas fuentes: mensajes del kernel, mensajes simples y estructurados del sistema, la salida estándar y error estándar de los servicios, así como los registros de auditoría del subsistema del kernel (para más detalles, consulte la página del manual de `systemd-journald`). Su misión es la de

crear y mantener un diario estructurado e indexado.

Su archivo de configuración es `/etc/systemd/journald.conf` y—como con cualquier otro servicio—puedes usar el comando `systemctl` para *iniciarlo*, *reiniciarlo*, *pararlo* o simplemente comprobar su *estado*:

```
root@debian:~# systemctl status systemd-journald
 systemd-journald.service - Journal Service
   Loaded: loaded (/lib/systemd/system/systemd-journald.service; static; vendor
preset: enabled)
   Active: active (running) since Sat 2019-10-12 13:43:06 CEST; 5min ago
     Docs: man:systemd-journald.service(8)
           man:journald.conf(5)
 Main PID: 178 (systemd-journal)
   Status: "Processing requests..."
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/systemd-journald.service
           └─178 /lib/systemd/systemd-journald
(...)
```

Los archivos de configuración del tipo `journal.conf.d/*.conf`—que pueden incluir configuraciones específicas de los paquetes—también son posibles (consulte la página del manual de `journald.conf` para saber más). Si se activa, el diario puede almacenarse de forma persistente en el disco o de forma volátil en un sistema de archivos basado en la memoria RAM. El diario no es un archivo de texto plano, es binario. Por lo tanto, no puede utilizar herramientas de análisis de texto como `less` o `more` para leer su contenido; en su lugar se utiliza el comando `journalctl`.

Consultando el contenido de journal

`journalctl` es la utilidad que se emplea para consultar el journal en `systemd`. Tienes que ser root o usar `sudo` para invocarlo. Si se consulta sin opciones, imprimirá todo el diario cronológicamente (con las entradas más antiguas listadas primero):

```
root@debian:~# journalctl
-- Logs begin at Sat 2019-10-12 13:43:06 CEST, end at Sat 2019-10-12 14:19:46 CEST.
--
Oct 12 13:43:06 debian kernel: Linux version 4.9.0-9-amd64 (debian-
kernel@lists.debian.org) (...)
Oct 12 13:43:06 debian kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.9.0-9-amd64
root=UUID=b6be6117-5226-4a8a-bade-2db35ccf4cf4 ro qu
(...)
```

Puede realizar consultas más específicas utilizando una serie de opciones:

`-r`

Los mensajes del journal se imprimirán en orden inverso:

```
root@debian:~# journalctl -r
-- Logs begin at Sat 2019-10-12 13:43:06 CEST, end at Sat 2019-10-12 14:30:30
CEST. --
Oct 12 14:30:30 debian sudo[1356]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)
Oct 12 14:30:30 debian sudo[1356]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/journalctl -r
Oct 12 14:19:53 debian sudo[1348]: pam_unix(sudo:session): session closed for
user root
(...)
```

`-f`

Imprimirá los mensajes más recientes del journal y seguirá imprimiendo las nuevas entradas a medida que se añadan a este, de forma similar a `tail -f`:

```
root@debian:~# journalctl -f
-- Logs begin at Sat 2019-10-12 13:43:06 CEST. --
(...)
Oct 12 14:44:42 debian sudo[1356]: pam_unix(sudo:session): session closed for
user root
Oct 12 14:44:44 debian sudo[1375]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/journalctl -f
Oct 12 14:44:44 debian sudo[1375]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)

(...)
```

`-e`

Saltará al final del journal para que las últimas entradas sean visibles dentro del localizador:

```
root@debian:~# journalctl -e
(...)
Oct 12 14:44:44 debian sudo[1375]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/journalctl -f
Oct 12 14:44:44 debian sudo[1375]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)
Oct 12 14:45:57 debian sudo[1375]: pam_unix(sudo:session): session closed for
user root
Oct 12 14:48:39 debian sudo[1378]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/journalctl -e
Oct 12 14:48:39 debian sudo[1378]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)
```

`-n <value>, --lines=<value>`

Imprimirá el *valor* de las líneas más recientes (si no se especifica `<valor>`, por defecto sera 10):

```
root@debian:~# journalctl -n 5
(...)
Oct 12 14:44:44 debian sudo[1375]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/journalctl -f
Oct 12 14:44:44 debian sudo[1375]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)
Oct 12 14:45:57 debian sudo[1375]: pam_unix(sudo:session): session closed for
user root
Oct 12 14:48:39 debian sudo[1378]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/journalctl -e
Oct 12 14:48:39 debian sudo[1378]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)
```

`-k`, `--dmesg`

Equivale a utilizar el comando `dmesg`:

```
root@debian:~# journalctl -k
-- Logs begin at Sat 2019-10-12 13:43:06 CEST, end at Sat 2019-10-12 14:53:20
CEST. --
Oct 12 13:43:06 debian kernel: Linux version 4.9.0-9-amd64 (debian-
kernel@lists.debian.org) (gcc version 6.3.0 20170516 (Debian 6.3.0-18
Oct 12 13:43:06 debian kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.9.0-9-
amd64 root=UUID=b6be6117-5226-4a8a-bade-2db35ccf4cf4 ro qu
Oct 12 13:43:06 debian kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87
floating point registers'
Oct 12 13:43:06 debian kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
registers'
(...)
```

Navegar y buscar a traves del Journal

Puede navegar por la salida de journal con:

- PageUp, PageDown y las teclas de flecha para moverse hacia arriba, abajo, izquierda y derecha.
- `>` Para ir al final de la salida.
- `<` Para ir al principio de la salida.

Puede buscar cadenas tanto hacia adelante como hacia atrás desde la posición de la vista actual:

- Búsqueda hacia delante: Pulse `/` e introduzca la cadena a buscar, luego pulse Enter.
- Búsqueda hacia atrás: Pulse `?` e introduzca la cadena a buscar, luego pulse Enter.

Para navegar por las coincidencias en las búsquedas, utilice `N` para ir a la siguiente ocurrencia y `Shift`+`N` para ir a la anterior.

Filtrar los datos de journal

El journal permite filtrar los datos del registro por diferentes criterios:

Número de arranque

`--list-boots`

Enumera todos los arranques disponibles. La salida consta de tres columnas; la primera especifica el número de arranque (`0` se refiere al arranque actual, `-1` es el anterior, `-2` el anterior al anterior y así sucesivamente); la segunda columna es el ID del arranque; la tercera muestra las marcas de tiempo:

```
root@debian:~# journalctl --list-boots
 0 83df3e8653474ea5aed19b41cdb45b78 Sat 2019-10-12 18:55:41 CEST—Sat 2019-10-
12 19:02:24 CEST
```

`-b`, `--boot`

Muestra todos los mensajes del arranque actual. Para ver los mensajes de registro de los arranques anteriores, sólo tiene que añadir un parámetro de desplazamiento como se ha explicado anteriormente. Por ejemplo, para que se impriman los mensajes del arranque anterior, escribirá `journalctl -b -1`. Recuerde, sin embargo, que para recuperar la información de los registros anteriores, la persistencia del diario debe estar habilitada (aprenderá cómo hacerlo en la siguiente sección):

```
root@debian:~# journalctl -b -1
La especificación del ID de arranque no tiene efecto, no se ha encontrado
ningún diario persistente.
```

Prioridad

`-p`

Curiosamente, también se puede filtrar por gravedad/prioridad con la opción `-p`:

```
root@debian:~# journalctl -b -0 -p err
-- No entries --
```

Journal nos informa que — hasta ahora — no ha habido ningún mensaje con una prioridad de `error` (o superior) desde el arranque actual. Nota: `-b -0` puede omitirse cuando se refiere al arranque actual.

NOTE Consulte la lección anterior para obtener una lista completa de todas las severidades (también conocidas como prioridades) de `syslog`.

Intervalo de tiempo

Puede hacer que `journalctl` imprima sólo los mensajes registrados dentro de un marco de tiempo específico utilizando las opciones `--since` y `--until`. La especificación de la fecha debe seguir el formato `AAAA-MM-DD HH:MM:SS`. Se asumirá que es medianoche si se omite el componente de tiempo. Del mismo modo, si se omite la fecha, se asume el día actual. Por ejemplo, para ver los mensajes registrados desde las 19:00 hasta las 19:01, se escribirá:

```
root@debian:~# journalctl --since "19:00:00" --until "19:01:00"
-- Logs begin at Sat 2019-10-12 18:55:41 CEST, end at Sat 2019-10-12 20:10:50
CEST. --
Oct 12 19:00:14 debian systemd[1]: Started Run anacron jobs.
Oct 12 19:00:14 debian anacron[1057]: Anacron 2.3 started on 2019-10-12
Oct 12 19:00:14 debian anacron[1057]: Normal exit (0 jobs run)
Oct 12 19:00:14 debian systemd[1]: anacron.timer: Adding 2min 47.988096s random
time.
```

Del mismo modo, puede utilizar una especificación de tiempo ligeramente diferente: `"integer time-unit ago"`. Por lo tanto, para ver los mensajes registrados hace dos minutos escribirá `sudo journalctl --since "2 minutes ago"`. También es posible utilizar `+` y `-` para especificar tiempos relativos a la hora actual, por lo que `--since "-2 minutos"` y `--since "2 minutes ago"` son

Además de las expresiones numéricas, puede especificar una serie de palabras clave:

`yesterday`
: A partir de la medianoche del día anterior al día actual.

`today`
: A partir de la medianoche del día actual.

`tomorrow`
: A partir de la medianoche del día siguiente al día actual.

`now`
: La hora actual.

Veamos todos los mensajes desde la pasada medianoche hasta hoy a las 21:00 horas:

```
root@debian:~# journalctl --since "today" --until "21:00:00"
-- Logs begin at Sat 2019-10-12 20:45:29 CEST, end at Sat 2019-10-12 21:06:15
CEST. --
Oct 12 20:45:29 debian sudo[1416]:    carol : TTY=pts/0 ; PWD=/home/carol ;
USER=root ; COMMAND=/bin/systemctl r
Oct 12 20:45:29 debian sudo[1416]: pam_unix(sudo:session): session opened for
user root by carol(uid=0)
Oct 12 20:45:29 debian systemd[1]: Stopped Flush Journal to Persistent
Storage.
(...)
```

NOTE Para saber más sobre las diferentes sintaxis de las especificaciones de tiempo, consulte la página del manual `systemd.time`.

Programa

Para ver los mensajes de journal relacionados con un ejecutable específico se utiliza la siguiente sintaxis `journalctl /ruta/al/ejecutable`:

```
root@debian:~# journalctl /usr/sbin/sshd
-- Logs begin at Sat 2019-10-12 20:45:29 CEST, end at Sat 2019-10-12 21:54:49
CEST. --
Oct 12 21:16:28 debian sshd[1569]: Accepted password for carol from 192.168.1.65
port 34050 ssh2
Oct 12 21:16:28 debian sshd[1569]: pam_unix(sshd:session): session opened for
user carol by (uid=0)
Oct 12 21:16:54 debian sshd[1590]: Accepted password for carol from 192.168.1.65
port 34052 ssh2
Oct 12 21:16:54 debian sshd[1590]: pam_unix(sshd:session): session opened for
user carol by (uid=0)
```

Unidad

Recuerda que una unidad es cualquier recurso manejado por `systemd` y también puedes filtrar por ellos.

`-u`

Muestra mensajes sobre una unidad específica:

```
root@debian:~# journalctl -u ssh.service
-- Logs begin at Sun 2019-10-13 10:50:59 CEST, end at Sun 2019-10-13 12:22:59
CEST. --
Oct 13 10:51:00 debian systemd[1]: Starting OpenBSD Secure Shell server...
Oct 13 10:51:00 debian sshd[409]: Server listening on 0.0.0.0 port 22.
Oct 13 10:51:00 debian sshd[409]: Server listening on :: port 22.
(...)
```

NOTE Para imprimir todas las unidades cargadas y activas, utilice `systemctl list-units`; para ver todos los archivos de unidad instalados utilice `systemctl list-unit-files`.

Campos

El journal también se puede filtrar por *campos* específicos mediante cualquiera de las siguientes sintaxis:

- `<field-name>=<value>`
- `_<field-name>=<value>_`
- `__<field-name>=<value>`

`PRIORITY=`

Uno de los ocho posibles valores de prioridad de `syslog` formateado como una cadena decimal:

```
root@debian:~# journalctl PRIORITY=3
-- Logs begin at Sun 2019-10-13 10:50:59 CEST, end at Sun 2019-10-13
14:30:50 CEST. --
Oct 13 10:51:00 debian avahi-daemon[314]: chroot.c: open() failed: No
such file or directory
```

Observe cómo podría haber conseguido la misma salida utilizando el comando `sudo journalctl -p err` que vimos anteriormente.

`SYSLOG_FACILITY=`

Cualquiera de los posibles números de código de instalación formateados como una cadena decimal. Por ejemplo, para ver todos los mensajes de nivel de usuario:

```
root@debian:~# journalctl SYSLOG_FACILITY=1
-- Logs begin at Sun 2019-10-13 10:50:59 CEST, end at Sun 2019-10-13
14:42:52 CEST. --
Oct 13 10:50:59 debian mtp-probe[227]: checking bus 1, device 2:
"/sys/devices/pci0000:00/0000:00:06.0/usb1/1-1"
Oct 13 10:50:59 debian mtp-probe[227]: bus: 1, device: 2 was not an MTP
device
Oct 13 10:50:59 debian mtp-probe[238]: checking bus 1, device 2:
"/sys/devices/pci0000:00/0000:00:06.0/usb1/1-1"
Oct 13 10:50:59 debian mtp-probe[238]: bus: 1, device: 2 was not an MTP
device
```

`_PID=`

Muestra los mensajes producidos por un ID de proceso específico. Para ver todos los

mensajes producidos por `systemd`, se debe escribir:

```
root@debian:~# journalctl _PID=1
-- Logs begin at Sun 2019-10-13 10:50:59 CEST, end at Sun 2019-10-13
14:50:15 CEST. --
Oct 13 10:50:59 debian systemd[1]: Mounted Debug File System.
Oct 13 10:50:59 debian systemd[1]: Mounted POSIX Message Queue File
System.
Oct 13 10:50:59 debian systemd[1]: Mounted Huge Pages File System.
Oct 13 10:50:59 debian systemd[1]: Started Remount Root and Kernel File
Systems.
Oct 13 10:50:59 debian systemd[1]: Starting Flush Journal to Persistent
Storage...
(...)
```

`_BOOT_ID`

Basándose en su ID de arranque, se pueden distinguir los mensajes de un arranque específico, por ejemplo: `sudo journalctl _BOOT_ID=83df3e8653474ea5aed19b41cdb45b78`.

`_TRANSPORT`

Mostrar los mensajes recibidos de un transporte específico. Los valores posibles son: `audit` (subsistema de auditoría del kernel), `driver` (generado internamente), `syslog` (socket syslog), `journal` (protocolo de diario nativo), `stdout` (salida estándar o error estándar de los servicios), `kernel` (buffer de anillo del kernel --lo mismo que `dmesg`, `journalctl -k` o `journalctl --dmesg`):

```
root@debian:~# journalctl _TRANSPORT=journal
-- Logs begin at Sun 2019-10-13 20:19:58 CEST, end at Sun 2019-10-13
20:46:36 CEST. --
Oct 13 20:19:58 debian systemd[1]: Started Create list of required static
device nodes for the current kernel.
Oct 13 20:19:58 debian systemd[1]: Starting Create Static Device Nodes in
/dev...
Oct 13 20:19:58 debian systemd[1]: Started Create Static Device Nodes in
/dev.
Oct 13 20:19:58 debian systemd[1]: Starting udev Kernel Device Manager...
(...)
```

Combinando campos

Los campos no son mutuamente excluyentes, por lo que puede utilizar más de uno en la misma consulta. Sin embargo, sólo se mostrarán los mensajes que coincidan con el valor de ambos campos simultáneamente:

```
root@debian:~# journalctl PRIORITY=3 SYSLOG_FACILITY=0
-- No entries --
root@debian:~# journalctl PRIORITY=4 SYSLOG_FACILITY=0
-- Logs begin at Sun 2019-10-13 20:19:58 CEST, end at Sun 2019-10-13 20:21:55 CEST.
--
Oct 13 20:19:58 debian kernel: acpi PNP0A03:00: fail to add MMCONFIG information,
can't access extended PCI configuration (...)
```

A menos que utilice el separador `+` para combinar dos expresiones a la manera de un *OR* lógico:

```
root@debian:~# journalctl PRIORITY=3 + SYSLOG_FACILITY=0
-- Logs begin at Sun 2019-10-13 20:19:58 CEST, end at Sun 2019-10-13 20:24:02 CEST.
--
Oct 13 20:19:58 debian kernel: Linux version 4.9.0-9-amd64 (debian-
kernel@lists.debian.org) (...9
Oct 13 20:19:58 debian kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.9.0-9-amd64
root=UUID= (...)
(...)
```

Por otro lado, puede proporcionar dos valores para el mismo campo y se mostrarán todas las entradas que coincidan con cualquiera de los dos valores:

```
root@debian:~# journalctl PRIORITY=1
-- Logs begin at Sun 2019-10-13 17:16:24 CEST, end at Sun 2019-10-13 17:30:14 CEST.
--
-- No entries --
root@debian:~# journalctl PRIORITY=1 PRIORITY=3
-- Logs begin at Sun 2019-10-13 17:16:24 CEST, end at Sun 2019-10-13 17:32:12 CEST.
--
Oct 13 17:16:27 debian connmand[459]: __connman_inet_get_pnp_nameservers: Cannot
read /pro
Oct 13 17:16:27 debian connmand[459]: The name net.connman.vpn was not provided by
any .se
```

NOTE Los campos de journal se encuentran en cualquiera de las siguientes categorías: “Campos de diario del usuario”, “Campos de diario de confianza”, “Campos de diario del núcleo”, “Campos en nombre de un programa diferente” y "Campos de dirección`". Para más información sobre este tema -incluyendo una lista completa de campos --vea la página `man` de `systemd.journal-fields(7)`.

Entradas manuales en el diario del sistema: `systemd-cat`

Al igual que el comando `logger` se utiliza para enviar mensajes desde la línea de comandos al registro del sistema (como vimos en la lección anterior), el comando `systemd-cat` tiene un propósito similar -pero más completo- con el diario del sistema. Nos permite enviar la entrada estándar (*stdin*), la salida (*stdout*) y el error (*stderr*) al diario.

Si se invoca sin parámetros, enviará todo lo que lea de *stdin* al diario. Una vez que haya terminado, pulse `Ctrl` + `C`:.

```
carol@debian:~$ systemd-cat
This line goes into the journal.
^C
```

Si se le pasa la salida de un comando canalizado, ésta se enviará también al diario:

```
carol@debian:~$ echo "And so does this line." | systemd-cat
```

Si va seguido de un comando, la salida de ese comando también se enviará al diario, junto con *stderr* (si lo hay):

```
carol@debian:~$ systemd-cat echo "And so does this line too."
```

También existe la posibilidad de especificar un nivel de prioridad con la opción `-p`:

```
carol@debian:~$ systemd-cat -p emerg echo "This is not a real emergency."
```

Consulte la página man de `systemd-cat` para conocer sus otras opciones.

Para ver las últimas cuatro líneas del diario utilice:

```
carol@debian:~$ journalctl -n 4
(...)
-- Logs begin at Sun 2019-10-20 13:43:54 CEST. --
Nov 13 23:14:39 debian cat[1997]: This line goes into the journal.
Nov 13 23:19:16 debian cat[2027]: And so does this line.
Nov 13 23:23:21 debian echo[2030]: And so does this line too.
Nov 13 23:26:48 debian echo[2034]: This is not a real emergency.
```

NOTE Las entradas con un nivel de prioridad de *emergencia* se imprimirán en negrita roja en la mayoría de los sistemas.

Almacenamiento persistente del diario

Como se ha mencionado anteriormente, tiene tres opciones en cuanto a la ubicación del diario:

- El registro en el diario puede ser desactivado por completo (aunque la redirección a otras instalaciones como la consola sigue siendo posible).
- Mantenerlo en memoria -lo que lo hace volátil- y deshacerse de los registros con cada reinicio del sistema. En este escenario, el directorio `/run/log/journal` será creado y utilizado.
- Hacerlo persistente para que escriba los registros en el disco. En este caso, los mensajes de registro irán al directorio `/var/log/journal`.

El comportamiento por defecto es el siguiente: si `/var/log/journal/` no existe, los registros se guardarán de forma volátil en un directorio en `/run/log/journal/` y — por tanto — se perderán al reiniciar. El nombre del directorio — el `/etc/machine-id` — es una cadena hexadecimal de 32 caracteres en minúsculas terminada en una nueva línea:

```
carol@debian:~$ ls /run/log/journal/8821e1fdf176445697223244d1dfbd73/
system.journal
```

Si intentas leerlo con less recibirás una advertencia, así que en su lugar utiliza el comando `journalctl`:

```
root@debian:~# less
/run/log/journal/9a32ba45ce44423a97d6397918de1fa5/system.journal
"/run/log/journal/9a32ba45ce44423a97d6397918de1fa5/system.journal" may be a binary
file.  See it anyway?
root@debian:~# journalctl
-- Logs begin at Sat 2019-10-05 21:26:38 CEST, end at Sat 2019-10-05 21:31:27 CEST.
--
(...)
Oct 05 21:26:44 debian systemd-journald[1712]: Runtime journal
(/run/log/journal/9a32ba45ce44423a97d6397918de1fa5) is 4.9M, max 39.5M, 34.6M free.
Oct 05 21:26:44 debian systemd[1]: Started Journal Service.
(...)
```

Si `/var/log/journal/` existe, los registros se almacenarán allí de forma persistente. Si se elimina este directorio, `systemd-journald` no lo recreará, sino que escribirá en `/run/log/journal`. Tan pronto como creemos `/var/log/journal/` de nuevo y reiniciemos el demonio, el registro persistente se restablecerá:

```
root@debian:~# mkdir /var/log/journal/
root@debian:~# systemctl restart systemd-journald
root@debian:~# journalctl
(...)
Oct 05 21:33:49 debian systemd-journald[1712]: Received SIGTERM from PID 1
(systemd).
Oct 05 21:33:49 debian systemd[1]: Stopped Journal Service.
Oct 05 21:33:49 debian systemd[1]: Starting Journal Service...
Oct 05 21:33:49 debian systemd-journald[1768]: Journal started
Oct 05 21:33:49 debian systemd-journald[1768]: System journal
(/var/log/journal/9a32ba45ce44423a97d6397918de1fa5) is 8.0M, max 1.1G, 1.1G free.
Oct 05 21:33:49 debian systemd[1]: Started Journal Service.
Oct 05 21:33:49 debian systemd[1]: Starting Flush Journal to Persistent Storage...
(...)
```

NOTE Por defecto, habrá archivos de diario específicos para cada usuario conectado, ubicados en `/var/log/journal/`, por lo que—junto con los archivos `system.journal`—también encontrará archivos del tipo `user-1000.journal`.

Además de lo que acabamos de mencionar, la forma en que el demonio del diario se ocupa del almacenamiento de registros puede cambiarse después de la instalación ajustando su archivo de configuración: `/etc/systemd/journald.conf`. La opción clave es `Storage=` y puede tener los siguientes valores:

`Storage=volatile`

Los datos del registro se almacenarán exclusivamente en la memoria — en `/run/log/journal/`. Si no está presente, se creará el directorio.

`Storage=persistent`

Por defecto, los datos de registro se almacenarán en el disco -en `/var/log/journal/`- con un retorno a la memoria (`/run/log/journal/`) durante las primeras etapas de arranque y si el disco no es escribible. Ambos directorios se crearán si es necesario.

`Storage=auto`

`auto` es similar a `persistent`, pero en este caso el directorio `/var/log/journal` no se creara si no fuese necesario. Esta es la opción por defecto.

`Storage=none`

Todos los datos de registro serán descartados. Sin embargo, el reenvío a otros objetivos como la consola, el buffer de registro del kernel o un socket syslog sigue siendo posible.

Por ejemplo, para que `systemd-journald` cree `/var/log/journal/` y cambie al almacenamiento persistente, debes editar `/etc/systemd/journald.conf` y establecer `Storage=persistent`, guardar el archivo y reiniciar el demonio con `sudo systemctl restart systemd-journald`. Para asegurarte de que el reinicio ha sido perfecto, siempre puedes comprobar el estado del demonio:

```
root@debian:~# systemctl status systemd-journald
 systemd-journald.service - Journal Service
   Loaded: loaded (/lib/systemd/system/systemd-journald.service; static; vendor
preset: enabled)
   Active: active (running) since Wed 2019-10-09 10:03:40 CEST; 2s ago
     Docs: man:systemd-journald.service(8)
           man:journald.conf(5)
 Main PID: 1872 (systemd-journal)
   Status: "Processing requests..."
    Tasks: 1 (limit: 3558)
   Memory: 1.1M
   CGroup: /system.slice/systemd-journald.service
           └─1872 /lib/systemd/systemd-journald

Oct 09 10:03:40 debian10 systemd-journald[1872]: Journal started
Oct 09 10:03:40 debian10 systemd-journald[1872]: System journal
(/var/log/journal/9a32ba45ce44423a97d6397918de1fa5) is 8.0M, max 1.2G, 1.2G free.
```

> **NOTE**
>
> Los archivos del diario en `/var/log/journal/<machine-id>/` o `/run/log/journal/<machine-id>/` tienen el sufijo `.journal` (por ejemplo, `system.journal`). Sin embargo, si se encuentra que están corruptos o el demonio se detiene de forma poco limpia, se renombrarán añadiendo `~` (por ejemplo, `system.journal~`) y el demonio empezará a escribir en un archivo nuevo y limpio.

Eliminación de datos antiguos del diario: Tamaño del diario

Los registros se guardan en *archivos de diario* cuyos nombres terminan en `.journal` o `.journal~` y se encuentran en el directorio apropiado (`/run/log/journal` o `/var/log/journal` según se haya configurado). Para comprobar cuánto espacio de disco ocupan actualmente los archivos de diario (tanto los archivados como los activos), utilice el parámetro `--disk-usage`:

```
root@debian:~# journalctl --disk-usage
Los diarios archivados y activos ocupan 24,0M en el sistema de archivos.
```

Los registros de `systemd` ocupan por defecto un máximo del 10% del tamaño del sistema de archivos donde se almacenan. Por ejemplo, en un sistema de archivos de 1GB no ocuparán más de 100MB. Una vez alcanzado este límite, los registros antiguos empezarán a desaparecer para mantenerse cerca de este valor.

Sin embargo, la aplicación del límite de tamaño de los archivos de diario almacenados puede gestionarse ajustando una serie de opciones de configuración en `/etc/systemd/journald.conf`. Estas opciones se dividen en dos categorías dependiendo del tipo de sistema de archivos utilizado: persistente (`/var/log/journal`) o en memoria (`/run/log/journal`). La primera utiliza opciones que llevan como prefijo la palabra `System` y sólo se aplicarán si el registro persistente está correctamente habilitado y una vez que el sistema haya arrancado por completo. Los nombres de las opciones de la segunda comienzan con la palabra `Runtime` y se aplicarán en los siguientes escenarios:

`SystemMaxUse=`, `RuntimeMaxUse=`

Controlan la cantidad de espacio en disco que puede ocupar el diario. Por defecto es el 10% del tamaño del sistema de archivos, pero puede modificarse (por ejemplo, `SystemMaxUse=500M`) siempre que no supere un máximo de 4GiB.

`SystemKeepFree=`, `RuntimeKeepFree=`

Controlan la cantidad de espacio en disco que debe quedar libre para otros usuarios. Por defecto es el 15% del tamaño del sistema de archivos, pero puede modificarse (por ejemplo, `SystemKeepFree=500M`) siempre que no supere un máximo de 4GiB.

En cuanto a la precedencia de `*MaxUse` y `*KeepFree`, `systemd-journald` satisfará ambos valores

utilizando el menor de los dos. Asimismo, tenga en cuenta que sólo se eliminan los ficheros de diario archivados (en contraposición a los activos).

`SystemMaxFileSize=`, `RuntimeMaxFileSize=`

Controlan el tamaño máximo al que pueden crecer los archivos individuales del diario. El valor por defecto es 1/8 de `*MaxUse`. La reducción de tamaño se realiza de forma sincrónica y los valores pueden especificarse en bytes o utilizando `K`, `M`, `G`, `T`, `P`, `E` para Kibibytes, Mebibytes, Gibibyte, Tebibytes, Pebibytes y Exbibytes, respectivamente.

`SystemMaxFiles=`, `RuntimeMaxFiles=`

Establecen el número máximo de ficheros de diario individuales y archivados a almacenar (los ficheros de diario activos no se ven afectados). El valor predeterminado es 100.

Además del borrado y la rotación de los mensajes de registro basados en el tamaño, `systemd-journald` también permite criterios basados en el tiempo utilizando las dos opciones siguientes: `MaxRetentionSec=` y `MaxFileSec=`. Consulte la página de manual de `journald.conf` para más información sobre estas y otras opciones.

NOTE Siempre que modifiques el comportamiento por defecto de `systemd-journald` descomentando y editando opciones en `/etc/systemd/journald.conf`, debes reiniciar el demonio para que los cambios surtan efecto.

Limpiando el Journal

Puede limpiar manualmente los ficheros de diario archivados en cualquier momento con cualquiera de las tres opciones siguientes:

`--vacuum-time=`

Esta opción basada en el tiempo eliminará todos los mensajes de los archivos de diario con una marca de tiempo más antigua que el marco temporal especificado. Los valores deben escribirse con cualquiera de los siguientes sufijos `s`, `m`, `h`, `days` (o `d`), `months`, `weeks` (o `w`) y `years` (o `y`). Por ejemplo, para eliminar todos los mensajes de los ficheros de diario archivados que tengan más de un mes de antigüedad:

```
root@debian:~# journalctl --vacuum-time=1months
Deleted archived journal
/var/log/journal/7203088f20394d9c8b252b64a0171e08/system@27dd08376f71405a91794e6
32ede97ed-0000000000000001-00059475764d46d6.journal (16.0M).
Deleted archived journal /var/log/journal/7203088f20394d9c8b252b64a0171e08/user-
1000@e7020d80d3af42f0bc31592b39647e9c-000000000000008e-00059479df9677c8.journal
(8.0M).
```

`--vacuum-size=`

Esta opción basada en el tamaño borrará los ficheros de diario archivados hasta que ocupen un valor inferior al tamaño especificado. Los valores deben escribirse con cualquiera de los siguientes sufijos: `K`, `M`, `G` o `T`. Por ejemplo, para eliminar los ficheros de diario archivados hasta que estén por debajo de 100 Mebibits:

```
root@debian:~# journalctl --vacuum-size=100M
Vacuuming done, freed 0B of archived journals from
/run/log/journal/9a32ba45ce44423a97d6397918de1fa5.
```

`--vacuum-files=`

Esta opción se encargará de que no queden más ficheros de diario archivados que el número especificado. El valor es un número entero. Por ejemplo, para limitar el número de ficheros de diario archivados a 10:

```
root@debian:~# journalctl --vacuum-files=10
Vacuuming done, freed 0B of archived journals from
/run/log/journal/9a32ba45ce44423a97d6397918de1fa5.
```

La aspiración sólo elimina los archivos del diario archivados. Si quiere deshacerse de todo (incluidos los archivos de diario activos), debe utilizar una señal (`SIGUSR2`) que solicite la rotación inmediata de los archivos de diario con la opción `--rotate`. Otras señales importantes pueden ser invocadas con las siguientes opciones:

`--flush` **(SIGUSR1)**

Solicita el volcado de los archivos del diario desde `/run/` a `/var/` para que el diario sea persistente. Requiere que el registro persistente esté habilitado y que `/var/` esté montado.

`--sync` **(SIGRTMIN+1)**

Se utiliza para solicitar que todos los datos de registro no escritos se escriban en el disco.

NOTE Para comprobar la consistencia interna del archivo del diario, utilice `journalctl` con la opción `--verify`. Verá una barra de progreso mientras se realiza la comprobación y se mostrará cualquier posible problema.

Recuperación de datos del diario de un sistema de rescate

Como administrador del sistema, puede encontrarse en una situación en la que necesite acceder a los archivos del diario en el disco duro de una máquina defectuosa a través de un sistema de rescate (un CD de arranque o una llave USB que contenga una distribución de Linux en vivo).

`journalctl` busca los archivos del diario en `/var/log/journal/<machine-id>/`. Debido a que los ID de las máquinas en los sistemas de rescate y en los sistemas defectuosos serán diferentes, debe utilizar la siguiente opción:

`-D </path/to/dir>`, `--directory=</path/to/dir>`

Con esta opción, especificamos una ruta de directorio donde `journalctl` buscará los archivos del diario en lugar de las ubicaciones por defecto del tiempo de ejecución y del sistema.

Por lo tanto, es necesario que monte el `rootfs` del sistema defectuoso (`/dev/sda1`) en el sistema de archivos en modo rescate y proceda a leer los archivos del diario así:

```
root@debian:~# journalctl -D /media/carol/faulty.system/var/log/journal/
-- Logs begin at Sun 2019-10-20 12:30:45 CEST, end at Sun 2019-10-20 12:32:57 CEST.
--
oct 20 12:30:45 suse-server kernel: Linux version 4.12.14-lp151.28.16-default
(geeko@buildhost) (...)
oct 20 12:30:45 suse-server kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.12.14-
lp151.28.16-default root=UUID=7570f67f-4a08-448e-aa09-168769cb9289 splash=>
oct 20 12:30:45 suse-server kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87
floating point registers'
oct 20 12:30:45 suse-server kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
registers'
(...)
```

Otras opciones que pueden ser útiles en este escenario son:

`-m`, `--merge`

Combina las entradas de todos los diarios disponibles en `/var/log/journal`, incluidos los remotos.

`--file`

Mostrará las entradas de un archivo específico, por ejemplo: `journalctl --file /var/log/journal/64319965bda04dfa81d3bc4e7919814a/user-1000.journal`.

`--root`

Se pasa como argumento una ruta de directorio que significa el directorio raíz. `journalctl` buscará allí los archivos del diario (por ejemplo, `journalctl --root /faulty.system/`).

Consulte la página de manual de `journalctl` para obtener más información.

Reenvío de datos de registro a un demonio `syslog` tradicional

Los datos de registro del diario pueden ponerse a disposición de un demonio `syslog` tradicional:

- Reenvío de mensajes al fichero socket `/run/systemd/journal/syslog` para que lo lea `syslogd`. Esta facilidad se activa con la opción `ForwardToSyslog=yes`.
- Tener un demonio `syslog` que se comporte como `journalctl`, por lo tanto leyendo los mensajes de registro directamente de los archivos del diario. En este caso, la opción relevante es la de `Storage`; debe tener un valor distinto de `none`.

NOTE

Asimismo, puedes reenviar los mensajes de registro a otros destinos con las siguientes opciones: `ForwardToKMsg` (buffer de registro del kernel — `kmsg`), `ForwardToConsole` (la consola del sistema) o `ForwardToWall` (todos los usuarios conectados a través de `wall`). Para más información, consulte la página man de `journald.conf`.

Ejercicios guiados

1. Asumiendo que eres `root`, completa la tabla con el comando `journalctl` apropiado:

Propósito	Comando
Imprimir entradas del kernel	
Imprimir los mensajes del segundo arranque empezando por el principio del diario	
Imprimir los mensajes del segundo arranque comenzando por el final del diario	
Imprimir los mensajes más recientes del diario y seguir vigilando los nuevos	
Imprime sólo los mensajes nuevos desde ahora, y actualiza la salida continuamente	
Imprime los mensajes del arranque anterior con prioridad de `advertencia` y en orden inverso	

2. El comportamiento del demonio del diario en relación con el almacenamiento está controlado principalmente por el valor de la opción `Storage` en `/etc/systemd/journald.conf`. Indique qué comportamiento está relacionado con qué valor en la siguiente tabla:

Comportamiento	`Almacenamiento=automático`	`Almacenamiento=ninguno`	`Almacenamiento=persistente`	`Almacenamiento=volátil`
Los datos del registro se desechan pero es posible el reenvío.				

Comportamiento	`Almacenamiento=automático`	`Almacenamiento=ninguno`	`Almacenamiento=persistente`	`Almacenamiento=volátil`
Una vez que el sistema ha arrancado, los datos de registro se almacenarán en `/var/log/journal`. Si no está presente, se creará el directorio.				
Una vez arrancado el sistema, los datos de registro se almacenarán en `/var/log/journal`. Si no está presente, el directorio no se creará.				
Los datos de registro se almacenarán en `/var/run/journal` pero no existiran después de los reinicios.				

3. Como ha aprendido, el diario se puede vaciar manualmente en función del tiempo, el tamaño y el número de archivos. Complete las siguientes tareas utilizando `journalctl` y las opciones apropiadas:

   - Compruebe cuánto espacio de disco ocupan los archivos del diario:

   ```
   ```

   - Reducir la cantidad de espacio reservado para los ficheros de diario archivados y fijarlo en 200MiB:

- Vuelva a comprobar el espacio en disco y explique los resultados:

Ejercicios de exploración

1. ¿Qué opciones deberías modificar en `/etc/systemd/journald.conf` para que los mensajes sean reenviados a `/dev/tty5`? ¿Qué valores deberían tener las opciones?

2. Proporcione el filtro correcto `journalctl` para imprimir lo siguiente:

Propósito	Filtro + Valor
Imprimir los mensajes de un usuario específico	
Imprimir mensajes de un host llamado `debian`	
Imprimir mensajes de un grupo específico	
Imprime los mensajes que pertenecen a `root`	
Basado en la ruta del ejecutable, imprime los mensajes de `sudo`	
Basado en el nombre del comando, imprime los mensajes de `sudo`	

3. Al filtrar por prioridad, los registros con una prioridad superior a la indicada también se incluirán en el listado; por ejemplo, el comando `journalctl -p err` imprimirá los mensajes de *error*, *crítico*, *alerta* y *emergencia*. Sin embargo, puedes hacer que `journalctl` muestre sólo un rango específico. ¿Qué comando usarías para que `journalctl` imprima sólo los mensajes de los niveles de prioridad *warning*, *error* y *critical*?

4. Los niveles de prioridad también se pueden especificar numéricamente. Vuelva a escribir el comando del ejercicio anterior utilizando la representación numérica de los niveles de prioridad:

Resumen

En esta lección aprendió:

- Las ventajas de utilizar `systemd` como gestor de sistemas y servicios.
- Los fundamentos de las unidades y objetivos de `systemd`.
- De donde `systemd-journald` obtiene los datos de registro.
- Las opciones que puedes pasar a `systemctl` para controlar `systemd-journald`: `start`, `status`, `restart` y `stop`.
- Donde se encuentra el archivo de configuración del diario — `/etc/systemd/journald.conf` — y sus principales opciones.
- Como consultar el diario de forma general y para datos específicos mediante el uso de filtros.
- Como navegar y buscar en el diario.
- Como tratar el almacenamiento de los archivos del diario: en memoria o en disco.
- Como desactivar el diario por completo.
- Como comprobar el espacio de disco ocupado por el diario, aplicar límites de tamaño a los archivos del diario almacenados y limpiar manualmente los archivos del diario archivados (*vacumming*).
- Como recuperar los datos del diario desde un sistema de rescate.
- Como reenviar los datos de registro a un demonio `syslog` tradicional.

Comandos usados en esta lección:

`systemctl`
: Controla el sistema `systemd` y el gestor de servicios.

`journalctl`
: Consultar el diario `systemd`.

`ls`
: Lista el contenido de un directorio.

`less`
: Ver el contenido del archivo.

`mkdir`

Hacer directorios.

Respuestas a los ejercicios guiados

1. Asumiendo que eres `root`, completa la tabla con el comando `journalctl` apropiado:

Propósito	Comando
Imprimir entradas del kernel	`journalctl -k` o `journalctl --dmesg`
Imprimir los mensajes del segundo arranque empezando por el principio del diario	`journalctl -b 2`
Imprimir los mensajes del segundo arranque comenzando por el final del diario	`journalctl -b -2 -r` or `journalctl -r -b -2`
Imprimir los mensajes más recientes del diario y seguir vigilando los nuevos	`journalctl -f`
Imprime sólo los mensajes nuevos desde ahora, y actualiza la salida continuamente	`journalctl --since "now" -f`
Imprime los mensajes del arranque anterior con prioridad de `advertencia` y en orden inverso	`journalctl -b -1 -p warning -r`

2. El comportamiento del demonio del diario en relación con el almacenamiento está controlado principalmente por el valor de la opción `Storage` en `/etc/systemd/journald.conf`. Indique qué comportamiento está relacionado con qué valor en la siguiente tabla:

Comportamiento	`Almacenamiento=automático`	`Almacenamiento=ninguno`	`Almacenamiento=persistente`	`Almacenamiento=volátil`
Los datos del registro se desechan pero es posible el reenvío.		x		

Comportamiento	`Almacenamiento=automático`	`Almacenamiento=ninguno`	`Almacenamiento=persistente`	`Almacenamiento=volátil`
Una vez que el sistema ha arrancado, los datos de registro se almacenarán en `/var/log/journal`. Si no está presente, se creará el directorio.			x	
Una vez arrancado el sistema, los datos de registro se almacenarán en `/var/log/journal`. Si no está presente, el directorio no se creará.	x			
Los datos de registro se almacenarán en `/var/run/journal` pero no existiran después de los reinicios.				x

3. Como ha aprendido, el diario se puede vaciar manualmente en función del tiempo, el tamaño y el número de archivos. Complete las siguientes tareas utilizando `journalctl` y las opciones apropiadas:

   ◦ Compruebe cuánto espacio de disco ocupan los archivos del diario:

```
journalctl --disk-usage
```

- Reducir la cantidad de espacio reservado para los ficheros de diario archivados y fijarlo en 200MiB:

```
journalctl --vacuum-size=200M
```

- Vuelve a comprobar el espacio en disco y explique los resultados:

```
journalctl --disk-usage
```

No hay correlación porque `--disk-usage` muestra el espacio ocupado tanto por los ficheros de diario activos como por los archivados, mientras que `--vacuum-size` sólo se aplica a los ficheros archivados.

Respuestas a los ejercicios de exploración

1. ¿Qué opciones deberías modificar en `/etc/systemd/journald.conf` para que los mensajes sean reenviados a `/dev/tty5`? ¿Qué valores deberían tener las opciones?

```
ForwardToConsole=yes
TTYPath=/dev/tty5
```

2. Proporcione el filtro correcto `journalctl` para imprimir lo siguiente:

Propósito	**Filtro + Valor**
Imprimir los mensajes de un usuario específico	`_ID=<user-id>`
Imprimir mensajes de un host llamado `debian`	`_HOSTNAME=debian`
Imprimir mensajes de un grupo específico	`_GID=<group-id>`
Imprime los mensajes que pertenecen a `root`	`_UID=0`
Basado en la ruta del ejecutable, imprime los mensajes de `sudo`	`_EXE=/usr/bin/sudo`
Basado en el nombre del comando, imprime los mensajes de `sudo`	`_COMM=sudo`

3. Al filtrar por prioridad, los registros con una prioridad superior a la indicada también se incluirán en el listado; por ejemplo, el comando `journalctl -p err` imprimirá los mensajes de *error*, *crítico*, *alerta* y *emergencia*. Sin embargo, puedes hacer que `journalctl` muestre sólo un rango específico. ¿Qué comando usarías para que `journalctl` imprima sólo los mensajes de los niveles de prioridad *warning*, *error* y *critical*?

```
journalctl -p warning..crit
```

4. Los niveles de prioridad también se pueden especificar numéricamente. Vuelva a escribir el comando del ejercicio anterior utilizando la representación numérica de los niveles de prioridad:

```
journalctl -p 4..2
```




108.3 Conceptos básicos del Agente de Transferencia de Correo

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 108.3

Importancia

3

Áreas de conocimiento clave

- Crear alias de correo electrónico.
- Configurar el reenvío de correo electrónico.
- Conocer los programas MTA más comunes (postfix, sendmail, qmail, exim) - sin incluir su configuración.

Lista parcial de archivos, términos y utilidades

- `~/.forward`
- sendmail emulation layer commands
- `newaliases`
- `mail`
- `mailq`
- `postfix`
- `sendmail`
- `exim`



108.3 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	108 Servicios esenciales del sistema
Objetivo:	108.3 Conceptos básicos del Agente de Transferencia de Correo
Lección:	1 de 1

Introducción

En los sistemas operativos tipo Unix, como Linux, cada usuario tiene su propia *bandeja de entrada*: una ubicación especial en el sistema de archivos a la que no pueden acceder otros usuarios que no sean root y que almacena los mensajes de correo electrónico personales del usuario. Los nuevos mensajes entrantes son añadidos a la bandeja de entrada del usuario por el *Mail Transfer Agent* (MTA). El MTA es un programa que se ejecuta como un servicio del sistema y que recoge los mensajes enviados por otras cuentas locales, así como los mensajes recibidos de la red, enviados desde cuentas de usuarios remotos.

El mismo MTA es también responsable de enviar mensajes a la red, si la dirección de destino se refiere a una cuenta remota. Para esto, utiliza una ubicación del sistema de archivos como *buzón de salida* de correo electrónico para todos los usuarios del sistema: tan pronto como un usuario coloque un nuevo mensaje en el buzón de salida, el MTA identificará el nodo de red de destino a partir del nombre de dominio dado por la dirección de correo electrónico de destino --después del signo `@`-- e intentará transferir el mensaje al MTA remoto utilizando el *Protocolo simple de transferencia de correo* (SMTP). SMTP se diseñó teniendo en cuenta las redes poco fiables, por lo que intentará establecer rutas de entrega alternativas si el nodo principal de destino del correo es inalcanzable.

MTA local y remoto

Las cuentas de usuario tradicionales en máquinas conectadas a la red constituyen el escenario de intercambio de correo electrónico más sencillo, en este cada, nodo de la red ejecuta su propio demonio MTA y no se requiere ningún otro software aparte del MTA para enviar y recibir mensajes de correo electrónico. En la práctica, sin embargo, es más común utilizar una cuenta de correo electrónico remota y no tener un servicio MTA local activo (es decir, utilizar una aplicación cliente de correo electrónico para acceder a la cuenta remota).

A diferencia de las cuentas locales, una cuenta de correo electrónico remota --también llamada *buzón remoto*-- requiere la autenticación del usuario para concederle acceso al buzón y al MTA remoto (en este caso, llamado simplemente *servidor SMTP*). Mientras que el usuario que interactúa con un buzón y un MTA locales ya está identificado por el sistema, un sistema remoto debe verificar la identidad del usuario antes de manejar sus mensajes a través de IMAP o POP3.

NOTE

Hoy en día, el método más común para enviar y recibir correo electrónico es a través de una cuenta alojada en un servidor remoto, por ejemplo, el servidor de correo electrónico centralizado de una empresa que aloja todas las cuentas de los empleados o un servicio de correo electrónico personal, como *Gmail* de Google. En lugar de recoger los mensajes entregados localmente, la aplicación cliente de correo electrónico se conectará al buzón remoto y recuperará los mensajes desde allí. Los protocolos POP3 e IMAP se utilizan habitualmente para recuperar los mensajes del servidor remoto, pero también pueden utilizarse otros protocolos propietarios no estándares.

Cuando se ejecuta un demonio MTA en el sistema local, los usuarios locales pueden enviar un correo electrónico a otros usuarios locales o a usuarios de una máquina remota, siempre que su sistema también tenga un servicio MTA que acepte conexiones de red. El puerto TCP 25 es el estándar para la comunicación SMTP, pero también se pueden utilizar otros puertos, dependiendo del esquema de autenticación y/o cifrado que se utilice (si lo hay).

Dejando de lado las topologías que implican el acceso a buzones remotos, se puede implementar una red de intercambio de correo electrónico entre cuentas de usuario ordinarias de Linux siempre que todos los nodos de la red tengan un MTA activo que sea capaz de realizar las siguientes tareas:

- Mantener la cola de salida de los mensajes a enviar. Para cada mensaje en cola, el MTA local evaluará el MTA de destino a partir de la dirección del destinatario.
- Comunicarse con demonios MTA remotos utilizando SMTP. El MTA local debe ser capaz de utilizar el Protocolo Simple de Transferencia de Correo (SMTP) a través de la pila TCP/IP para recibir, enviar y redirigir mensajes de/a otros demonios MTA remotos.

- Mantener una bandeja de entrada individual para cada cuenta local. El MTA suele almacenar los mensajes en el formato *mbox*: que es un único archivo de texto que contiene todos los mensajes de correo electrónico en secuencia.

Normalmente, las direcciones de correo electrónico especifican un nombre de dominio como ubicación, por ejemplo, `lpi.org` en `info@lpi.org`. En este caso, el MTA del remitente consultará al servicio DNS el registro MX correspondiente. El registro DNS MX contiene la dirección IP del MTA que gestiona el correo electrónico para ese dominio. Si el mismo dominio tiene más de un registro MX especificado en el DNS, el MTA debe intentar ponerse en contacto con ellos según sus valores de prioridad. Si la dirección del destinatario no especifica un nombre de dominio o el dominio no tiene un registro MX, la parte que sigue al símbolo `@` se tratará como el host del MTA de destino.

Hay que tener en cuenta los aspectos de seguridad si los hosts del MTA van a ser visibles para los hosts de Internet. Por ejemplo, es posible que un usuario desconocido utilice el MTA local para hacerse pasar por otro usuario y enviar correos electrónicos potencialmente dañinos. Un MTA que retransmite ciegamente un correo electrónico se conoce como *retransmisión abierta*, cuando puede ser utilizado como intermediario para disfrazar potencialmente el verdadero remitente del mensaje. Para evitar estos usos indebidos, la recomendación es aceptar conexiones sólo de dominios autorizados e implementar un esquema de autenticación seguro.

Además, hay muchas implementaciones diferentes de MTA para Linux, cada una de las cuales se centra en aspectos específicos como la compatibilidad, el rendimiento, la seguridad, etc. Sin embargo, todos los MTAs seguirán los mismos principios básicos y proporcionarán características similares.

MTAs de Linux

El MTA tradicional disponible para los sistemas Linux es *Sendmail*, un MTA de propósito general muy flexible utilizado por muchos sistemas operativos tipo Unix. Otros MTA comunes son *Postfix*, *qmail* y *Exim*. La razón principal para elegir un MTA alternativo es implementar características avanzadas más fácilmente, ya que configurar servidores de correo electrónico personalizados en Sendmail puede ser una tarea complicada. Además, cada distribución puede tener su MTA preferido, con configuraciones predefinidas. Todos los MTA pretenden ser sustitutos de Sendmail, por lo que todas las aplicaciones compatibles con Sendmail deberían funcionar independientemente del MTA que se utilice.

Si el MTA está funcionando pero no acepta conexiones de red, sólo podrá entregar mensajes de correo electrónico en la máquina local. Para el MTA `sendmail`, el archivo `/etc/mail/sendmail.mc` debe ser modificado para aceptar conexiones no locales. Para ello, la entrada sera la siguiente:

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

Debe modificarse a la dirección de red correcta y el servicio debe reiniciarse. Algunas distribuciones de Linux, como Debian, pueden ofrecer herramientas de configuración para ayudar a poner en marcha el servidor de correo electrónico con un conjunto predefinido de características de uso común.

> **TIP** Debido a problemas de seguridad, la mayoría de las distribuciones de Linux no instalan un MTA por defecto. Para probar los ejemplos dados en esta lección, asegúrese de que hay un MTA en funcionamiento en cada máquina y que están aceptando conexiones en el puerto TCP 25. Por razones de seguridad, estos sistemas no deberían estar expuestos a conexiones entrantes de la Internet pública durante las pruebas.

Una vez que el MTA está funcionando y aceptando conexiones de la red, los nuevos mensajes de correo electrónico se le pasan comandos SMTP que se envían a través de una conexión TCP. El comando `nc` — una utilidad de red que lee y escribe datos genéricos a través de la red — puede utilizarse para enviar comandos SMTP directamente al MTA. Si el comando `nc` no está disponible, se instalará con el paquete *ncat* o *nmap-ncat*, dependiendo del sistema de gestión de paquetes que se utilice. Escribir comandos SMTP directamente al MTA le ayudará a entender mejor el protocolo y otros conceptos generales del correo electrónico, pero también puede ayudar a diagnosticar problemas en el proceso de entrega del correo.

Si, por ejemplo, el usuario `emma` en el host `lab1.campus` quiere enviar un mensaje al usuario `dave` en el host `lab2.campus`, entonces puede usar el comando `nc` para conectarse directamente al MTA `lab2.campus`, asumiendo que está escuchando en el puerto TCP 25:

```
$ nc lab2.campus 25
220 lab2.campus ESMTP Sendmail 8.15.2/8.15.2; Sat, 16 Nov 2019 00:16:07 GMT
HELO lab1.campus
250 lab2.campus Hello lab1.campus [10.0.3.134], pleased to meet you
MAIL FROM: emma@lab1.campus
250 2.1.0 emma@lab1.campus... Sender ok
RCPT TO: dave@lab2.campus
250 2.1.5 dave@lab2.campus... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Subject: Recipient MTA Test

Hi Dave, this is a test for your MTA.
.
250 2.0.0 xAG0G7Y0000595 Message accepted for delivery
QUIT
221 2.0.0 lab2.campus closing connection
```

Una vez establecida la conexión, el MTA remoto se identifica y está listo para recibir comandos SMTP. El primer comando SMTP del ejemplo, `HELO lab1.campus`, indica que `lab1.campus` es el iniciador del intercambio. Los dos siguientes comandos, `MAIL FROM: emma@lab1.campus` y `RCPT TO: dave@lab2.campus`, indican el remitente y el destinatario. El mensaje de correo electrónico propiamente dicho comienza después del comando `DATA` y termina con un punto en una línea por sí mismo. Para añadir un campo `subject` al correo electrónico, debe estar en la primera línea después del comando `DATA`, como se muestra en el ejemplo. Cuando se utiliza el campo de asunto, debe haber una línea vacía que lo separe del contenido del correo electrónico. El comando `QUIT` termina la conexión con el MTA en el host `lab2.campus`.

En el host `lab2.campus`, el usuario `dave` recibirá un mensaje similar a `You have new mail in /var/spool/mail/dave` tan pronto como entre en una sesión de shell. Este archivo contendrá el mensaje de correo electrónico en bruto enviado por `emma` así como las cabeceras añadidas por el MTA:

```
$ cat /var/spool/mail/dave
From emma@lab1.campus  Sat Nov 16 00:19:13 2019
Return-Path: <emma@lab1.campus>
Received: from lab1.campus (lab1.campus [10.0.3.134])
    by lab2.campus (8.15.2/8.15.2) with SMTP id xAG0G7Y0000595
    for dave@lab2.campus; Sat, 16 Nov 2019 00:17:06 GMT
Date: Sat, 16 Nov 2019 00:16:07 GMT
From: emma@lab1.campus
Message-Id: <201911160017.xAG0G7Y0000595@lab2.campus>
Subject: Recipient MTA Test

Hi Dave, this is a test for your MTA.
```

La cabecera `Received:` muestra que el mensaje de `lab1.campus` fue recibido directamente por `lab2.campus`. Por defecto, los MTAs sólo aceptan mensajes a destinatarios locales. El siguiente error probablemente ocurrirá si el usuario `emma` intenta enviar un correo electrónico al usuario `henry` en el host `lab3.campus`, pero utilizando el MTA `lab2.campus` en lugar del MTA apropiado `lab3.campus`:

```
$ nc lab2.campus 25
220 lab2.campus ESMTP Sendmail 8.15.2/8.15.2; Sat, 16 Nov 2019 00:31:44 GMT
HELO lab1.campus
250 lab2.campus Hello lab1.campus [10.0.3.134], pleased to meet you
MAIL FROM: emma@lab1.campus
250 2.1.0 emma@lab1.campus... Sender ok
RCPT TO: henry@lab3.campus
550 5.7.1 henry@lab3.campus... Relaying denied
```

Los números de respuesta SMTP que empiezan por 5, como el mensaje `Relaying denied`, indican un error. Hay situaciones legítimas en las que la retransmisión es deseable, como cuando los hosts que envían y reciben correos electrónicos no están conectados todo el tiempo: se puede configurar un MTA intermedio para que acepte los correos electrónicos destinados a otros hosts, actuando como un servidor SMTP *de retransmisión* que puede reenviar mensajes entre MTAs.

La posibilidad de enrutar el tráfico de correo electrónico a través de servidores SMTP intermedios desaconseja el intento de conectarse directamente al host indicado por la dirección de correo electrónico del destinatario, como se muestra en los ejemplos anteriores. Además, las direcciones de correo electrónico suelen tener un nombre de dominio como ubicación (después de la `@`), por lo que el nombre real del host MTA correspondiente debe ser recuperado a través de DNS. Por lo tanto, se recomienda delegar la tarea de identificar el host de destino apropiado al MTA local o al servidor SMTP remoto, cuando se utilizan buzones remotos.

Sendmail proporciona el comando `sendmail` para realizar muchas operaciones relacionadas con el correo electrónico, incluyendo la asistencia en la composición de nuevos mensajes. También requiere que el usuario escriba las cabeceras del correo electrónico a mano, pero de una forma más amigable que utilizando los comandos SMTP directamente. Así que un método más adecuado para que el usuario `emma@lab1.campus` envíe un mensaje de correo electrónico a `dave@lab2.campus` sería:

```
$ sendmail dave@lab2.campus
From: emma@lab1.campus
To: dave@lab2.campus
Subject: Sender MTA Test

Hi Dave, this is a test for my MTA.
.
```

También en este caso, el punto en una línea por sí mismo termina el mensaje. El mensaje debería ser enviado inmediatamente al destinatario, a menos que el MTA local no se haya podido contactar con el MTA remoto. El comando `mailq`, si es ejecutado por root, mostrará todos los mensajes no entregados. Si, por ejemplo, el MTA de `lab2.campus` no ha respondido, el comando `mailq` mostrará el mensaje no entregado y la causa del fallo:

```
# mailq
        /var/spool/mqueue (1 request)
-----Q-ID----- --Size-- -----Q-Time----- ------------Sender/Recipient-----------
xAIK3D9S000453       36 Mon Nov 18 20:03 <emma@lab1.campus>
                 (Deferred: Connection refused by lab2.campus.)
                     <dave@lab2.campus>
        Total requests: 1
```

La ubicación por defecto de la cola de salida es `/var/spool/mqueue/`, pero diferentes MTAs pueden utilizar diferentes ubicaciones en el directorio `/var/spool/`. Postfix, por ejemplo, creará un árbol de directorios en `/var/spool/postfix/` para gestionar la cola. El comando `mailq` es equivalente a `sendmail -bp`, debe estar presente independientemente del MTA instalado en el sistema. Para asegurar la compatibilidad con versiones anteriores, la mayoría de los MTAs proporcionan estos comandos tradicionales de administración de correo.

Si el host principal de destino del correo electrónico — cuando se proporciona desde un registro DNS MX para el dominio — es inalcanzable, el MTA intentará ponerse en contacto con las entradas con menor prioridad (si hay alguna especificada). Si ninguna de ellas es alcanzable, el mensaje permanecerá en la cola de la bandeja de salida local para ser enviado más tarde. Si se configura para ello, el MTA puede comprobar periódicamente la disponibilidad de los hosts remotos y realizar un

nuevo intento de entrega. Si se utiliza un MTA compatible con Sendmail, se realizará inmediatamente un nuevo intento con el comando `sendmail -q`.

Sendmail almacenará los mensajes entrantes en un archivo con el nombre del propietario de la bandeja de entrada correspondiente, por ejemplo `/var/spool/mail/dave`. Otros MTA, como Postfix, pueden almacenar los mensajes de correo electrónico entrantes en ubicaciones como `/var/mail/dave`, pero el contenido del archivo es el mismo. En el ejemplo, el comando `sendmail` se utilizó en el host del remitente para crear el mensaje, por lo que las cabeceras de los mensajes sin procesar muestran que el correo electrónico siguió pasos adicionales antes de llegar al destino final:

```
$ cat /var/spool/mail/dave
From emma@lab1.campus  Mon Nov 18 20:07:39 2019
Return-Path: <emma@lab1.campus>
Received: from lab1.campus (lab1.campus [10.0.3.134])
    by lab2.campus (8.15.2/8.15.2) with ESMTPS id xAIK7clC000432
    (version=TLSv1.3 cipher=TLS_AES_256_GCM_SHA384 bits=256 verify=NOT)
    for <dave@lab2.campus>; Mon, 18 Nov 2019 20:07:38 GMT
Received: from lab1.campus (localhost [127.0.0.1])
    by lab1.campus (8.15.2/8.15.2) with ESMTPS id xAIK3D9S000453
    (version=TLSv1.3 cipher=TLS_AES_256_GCM_SHA384 bits=256 verify=NOT)
    for <dave@lab2.campus>; Mon, 18 Nov 2019 20:03:13 GMT
Received: (from emma@localhost)
    by lab1.campus (8.15.2/8.15.2/Submit) id xAIK0doL000449
    for dave@lab2.campus; Mon, 18 Nov 2019 20:00:39 GMT
Date: Mon, 18 Nov 2019 20:00:39 GMT
Message-Id: <201911182000.xAIK0doL000449@lab1.campus>
From: emma@lab1.campus
To: dave@lab2.campus
Subject: Sender MTA Test

Hi Dave, this is a test for my MTA.
```

De abajo a arriba, las líneas que comienzan con `Received:` muestran la ruta seguida por el mensaje. El mensaje fue enviado por el usuario `emma` con el comando `sendmail dave@lab2.campus` emitido en `lab1.campus`, como se indica en la primera cabecera `Received:`. A continuación, todavía en `lab1.campus`, el MTA utiliza ESMTPS — un superconjunto del SMTP, que añade extensiones de cifrado — para enviar el mensaje al MTA en `lab2.campus`, como se indica en la última cabecera (superior) `Received:`.

El MTA termina su trabajo una vez que el mensaje se guarda en la bandeja de entrada del usuario. Es habitual realizar algún tipo de filtrado del correo electrónico, como el bloqueo de spam o la aplicación de reglas de filtrado definidas por el usuario. Estas tareas son ejecutadas por aplicaciones

de terceros, que trabajan conjuntamente con el MTA. El MTA podría, por ejemplo, llamar a la utilidad *SpamAssassin* para marcar los mensajes sospechosos utilizando sus funciones de análisis de texto.

Aunque es posible, no es conveniente leer el archivo del buzón directamente. Se recomienda utilizar en su lugar un programa cliente de correo electrónico (por ejemplo, Thunderbird, Evolution o KMail), que analizará el archivo y gestionará adecuadamente los mensajes. Estos programas también ofrecen funciones adicionales, como accesos directos a acciones comunes, subdirectorios de la bandeja de entrada, etc.

El comando `mail` y los agentes de usuario de correo (MUA)

Es posible escribir un mensaje de correo electrónico directamente en su formato crudo, pero es mucho más práctico utilizar una aplicación cliente — también conocida como MUA (*Agente de Usuario de Correo*) — para acelerar el proceso y evitar errores. El MUA se encarga del trabajo bajo el capó, es decir, el cliente de correo electrónico presenta y organiza los mensajes recibidos y maneja la comunicación adecuada con el MTA después de que el usuario compone un correo electrónico.

Hay muchos tipos distintos de agentes de usuario de correo. Las aplicaciones de escritorio como *Mozilla Thunderbird* y *Evolution* de Gnome soportan cuentas de correo electrónico tanto locales como remotas. Incluso las interfaces de *Webmail* pueden considerarse un tipo de MUA, ya que interviene la interacción entre el usuario y el MTA subyacente. Sin embargo, los clientes de correo electrónico no se limitan a las interfaces gráficas. Los clientes de correo electrónico de consola se utilizan ampliamente para acceder a los buzones no integrados en una interfaz gráfica y para automatizar las tareas relacionadas con el correo electrónico dentro de los scripts del shell.

Originalmente, el comando `mail` de Unix sólo estaba pensado para compartir mensajes entre usuarios del sistema local (el primer comando `mail` se remonta a la primera edición de Unix, lanzada en 1971). Cuando los intercambios de correo electrónico en red se hicieron más importantes, se crearon otros programas para hacer frente al nuevo sistema de entrega y sustituyeron gradualmente al antiguo programa `mail`.

Hoy en día, el comando `mail` más utilizado es el proporcionado por el paquete *mailx*, que es compatible con todas las características modernas del correo electrónico. En la mayoría de las distribuciones de Linux, el comando `mail` es sólo un enlace simbólico al comando `mailx`. Otras implementaciones, como el paquete *GNU Mailutils*, proporcionan básicamente las mismas características que `mailx`. Sin embargo, existen ligeras diferencias entre ellas, especialmente en lo que respecta a las opciones de la línea de comandos.

Independientemente de su implementación, todas las variantes modernas del comando `mail` funcionan en dos modos: *modo normal* y *modo de envío*. Si se proporciona una dirección de correo

electrónico como argumento al comando `mail`, éste entrará en el modo de envío, de lo contrario entrará en el modo normal (lectura). En el modo normal, los mensajes recibidos se listan con un índice numérico para cada uno, de modo que el usuario puede referirse a ellos individualmente cuando escribe comandos en el prompt interactivo. El comando `print 1` puede utilizarse para mostrar el contenido del mensaje número 1, por ejemplo. Los comandos interactivos pueden ser abreviados, por lo que comandos como `print`, `delete` o `reply` pueden ser reemplazados por `p`, `d` o `r`, respectivamente. El comando `mail` siempre considerará el último mensaje recibido o el último visto cuando se omita el número de índice del mensaje. El comando `quit` o `q` terminará del programa.

El modo *send mode* es especialmente útil para enviar mensajes de correo electrónico automatizados. Puede utilizarse, para enviar un correo electrónico al administrador del sistema si un script de mantenimiento programado no realiza su tarea. En el modo de envío, `mail` utilizará el contenido de la *entrada estándar* como cuerpo del mensaje:

```
$ mail -s "Maintenance fail" henry@lab3.campus <<<"The maintenance script failed at
`date`"
```

En este ejemplo, se añadió la opción `-s` para incluir un campo de asunto al mensaje. El cuerpo del mensaje fue proporcionado por la redirección *Hereline* a la entrada estándar, pero el contenido de un archivo o la salida de un comando también podría ser canalizado a la *stdin* del programa. Si no se proporciona ningún contenido mediante una redirección a la entrada estándar, el programa esperará a que el usuario introduzca el cuerpo del mensaje. En este caso, la pulsación de la tecla Ctrl + D finalizará el mensaje. El comando `mail` saldrá inmediatamente después de que el mensaje se añada a la cola de salida.

Personalización de la entrega

Por defecto, las cuentas de correo electrónico en un sistema Linux están asociadas a las cuentas estándares del sistema. Por ejemplo, si el usuario Carol tiene el nombre de usuario `carol` en el host `lab2.campus` entonces su dirección de correo electrónico será `carol@lab2.campus`. Esta asociación uno a uno entre las cuentas del sistema y los buzones de correo puede ser extendida por métodos estándares proporcionados por la mayoría de las distribuciones de Linux, en particular el mecanismo de enrutamiento de correo electrónico proporcionado por el archivo `/etc/aliases`.

Un alias de correo electrónico es un destinatario de correo electrónico “virtual” cuyos mensajes recibidos se redirigen a buzones locales existentes o a otros tipos de destinos de almacenamiento o procesamiento de mensajes. Los alias son útiles, por ejemplo, para colocar los mensajes enviados a `postmaster@lab2.campus` en el buzón de Carol, que es un buzón local ordinario en el sistema `lab2.campus`. Para ello, se debe añadir la línea `postmaster: carol` al fichero `/etc/aliases` en `lab2.campus`. Después de modificar el archivo `/etc/aliases`, se debe ejecutar el comando

`newaliases` para actualizar la base de datos de alias del MTA y hacer efectivos los cambios. Los comandos `sendmail -bi` o `sendmail -I` también pueden utilizarse para actualizar la base de datos de alias. Los alias se definen por línea, con el formato `<alias>: <destino>`. Además de los buzones locales ordinarios, indicados por el nombre de usuario correspondiente, existen otros tipos de destino:

- Una ruta completa (que comienza con `/`) a un archivo. Los mensajes enviados al alias correspondiente se añadirán al archivo.
- Un comando para procesar el mensaje. El `<destino>` debe comenzar con un caracter de tubería y si el comando contiene caracteres especiales (como espacios en blanco), debe ir entre comillas dobles. Por ejemplo, el alias `subscribe: |subscribe.sh` en `lab2.campus` reenviará todos los mensajes enviados a `subscribe@lab2.campus` a la entrada estándar del comando `subscribe.sh`. Si sendmail se ejecuta en *modo shell restringido*, los comandos permitidos — o los enlaces a ellos — deben estar en `/etc/smrsh/`.
- Un archivo de inclusión. Un solo alias puede tener múltiples destinos (separados por comas), por lo que puede ser más práctico mantenerlos en un archivo externo. La palabra clave `:include:` debe indicar la ruta del archivo, como en `:include:/var/local/destinos`.
- Una dirección externa. Los alias también pueden reenviar mensajes a direcciones de correo electrónico externas.
- Otro alias.

Un usuario local sin privilegios puede definir alias para su propio correo electrónico editando el archivo `.forward` en su directorio personal. Como los alias sólo pueden afectar a su propio buzón, sólo es necesaria la parte de `<destination>`. Para reenviar todos los correos electrónicos entrantes a una dirección externa, por ejemplo, el usuario `dave` en `lab2.campus` podría crear el siguiente archivo `~/.forward`:

```
$ cat ~/.forward
emma@lab1.campus
```

Reenviará todos los mensajes de correo electrónico enviados a `dave@lab2.campus` a `emma@lab1.campus`. Al igual que con el fichero `/etc/aliases`, se pueden añadir (una por línea) otras reglas de redirección a `.forward`. Sin embargo, el archivo `.forward` debe ser escribible sólo por su propietario y no es necesario ejecutar el comando `newaliases` después de modificarlo. Los archivos que comienzan con un punto no aparecen en los listados de archivos habituales, lo que podría hacer que el usuario no conociera los alias activos. Por lo tanto, es importante verificar si el archivo existe cuando se diagnostican problemas de entrega de correo electrónico.

Ejercicios guiados

1. Sin más opciones o argumentos, el comando `mail henry@lab3.campus` entra en el modo de entrada para que el usuario pueda escribir el mensaje a `henry@lab3.campus`. Después de terminar el mensaje, ¿qué tecla cerrará el modo de entrada y enviará el correo electrónico?

2. ¿Qué comando puede ejecutar el usuario root para listar los mensajes no entregados que se originaron en el sistema local?

3. ¿Cómo puede un usuario sin privilegios utilizar el método MTA estándar para reenviar automáticamente todo su correo entrante a la dirección `dave@lab2.campus`?

Ejercicios de exploración

1. Utilizando el comando `mail` proporcionado por `mailx`, ¿qué comando enviará un mensaje a `emma@lab1.campus` con el archivo `logs.tar.gz` como adjunto y la salida del comando `uname -a` como cuerpo del correo electrónico?

2. Un administrador de servicios de correo electrónico quiere supervisar las transferencias de correo electrónico a través de la red, pero no quiere saturar su buzón con mensajes de prueba. ¿Cómo podría este administrador configurar un alias de correo electrónico en todo el sistema para redirigir todo el correo electrónico enviado al usuario `test` al archivo `/dev/null`?

3. ¿Qué comando, además de `newaliases`, podría utilizarse para actualizar la base de datos de alias después de añadir un nuevo alias a `/etc/aliases`?

Resumen

Esta lección cubre el papel y el uso de los Agentes de Transferencia de Correo en los sistemas Linux. El MTA proporciona un método estándar para la comunicación entre cuentas de usuario y puede combinarse con otro software para proporcionar una funcionalidad adicional. La lección trató los siguientes temas:

- Conceptos sobre tecnologías, buzones y protocolos relacionados con el correo electrónico.
- Cómo los MTAs de Linux intercambian mensajes a través de la red.
- Clientes de correo electrónico de consola y MUAs (Agentes de Usuario de Correo).
- Uso de alias y reenvío de correo electrónico local.

En esta lección fueron abordados las siguientes tecnologías, comandos y procedimientos:

- SMTP y protocolos relacionados.
- MTAs disponibles para Linux: Sendmail, Postfix, qmail, Exim.
- Comandos MTA y MUA: `sendmail` y `mail`.
- Archivos y comandos administrativos: `mailq`, `/etc/aliases`, `newaliases`, `~/.forward`.

Respuestas a los ejercicios guiados

1. Sin más opciones o argumentos, el comando `mail henry@lab3.campus` entra en el modo de entrada para que el usuario pueda escribir el mensaje a `henry@lab3.campus`. Después de terminar el mensaje, ¿qué tecla cerrará el modo de entrada y enviará el correo electrónico?

   Al pulsar `Ctrl` + `D` el programa se cerrará y enviará el correo electrónico.

2. ¿Qué comando puede ejecutar el usuario root para listar los mensajes no entregados que se originaron en el sistema local?

   El comando `mailq` o `sendmail -bp`.

3. ¿Cómo puede un usuario sin privilegios utilizar el método MTA estándar para reenviar automáticamente todo su correo entrante a la dirección `dave@lab2.campus`?

   El usuario debe añadir `dave@lab2.campus` a `~/.forward`.

Respuestas a los ejercicios de exploración

1. Utilizando el comando `mail` proporcionado por `mailx`, ¿qué comando enviará un mensaje a `emma@lab1.campus` con el archivo `logs.tar.gz` como adjunto y la salida del comando `uname -a` como cuerpo del correo electrónico?

   ```
   uname -a | mail -a logs.tar.gz emma@lab1.campus
   ```

2. Un administrador de servicios de correo electrónico quiere supervisar las transferencias de correo electrónico a través de la red, pero no quiere saturar su buzón con mensajes de prueba. ¿Cómo podría este administrador configurar un alias de correo electrónico en todo el sistema para redirigir todo el correo electrónico enviado al usuario `test` al archivo `/dev/null`?

   La línea `test: /dev/null` en `/etc/aliases` redirigirá todos los mensajes enviados al buzón local `test` al archivo `/dev/null`.

3. ¿Qué comando, además de `newaliases`, podría utilizarse para actualizar la base de datos de alias después de añadir un nuevo alias a `/etc/aliases`?

   Comando `sendmail -bi` o `sendmail -I`.



108.4 Gestión de la impresión y de las impresoras

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 108.4

Importancia

2

Áreas de conocimiento clave

- Configuración básica de CUPS (para impresoras locales y remotas).
- Gestión de colas de impresión de los usuarios.
- Resolución de problemas de impresión.
- Agregar y eliminar trabajos en colas de impresión configuradas.

Lista parcial de archivos, términos y utilidades

- Utilidades, herramientas y archivos de configuración de CUPS
- `/etc/cups/`
- Interfaz legacy de lpd (`lpr`, `lprm`, `lpq`)



108.4 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	108 Servicios esenciales del sistema
Objetivo:	108.4 Gestión de la impresión y de las impresoras
Lección:	1 de 1

Introducción

Las declaraciones de una "sociedad sin papel", provocadas por la llegada de las computadoras, han demostrado ser falsas hasta la fecha. Muchas organizaciones siguen confiando en las páginas impresas o en la "hard copy" de la información. Teniendo esto en cuenta, podemos ver lo importante que es para un usuario saber cómo imprimir desde un sistema, así como para un administrador que necesita saber cómo mantener la capacidad de un servidor para trabajar con impresoras.

En Linux, así como en muchos otros sistemas operativos, la pila de software *Common Unix Printing System* (CUPS) permite imprimir y gestionar las impresoras desde un equipo. A continuación se muestra un esquema muy simplificado de cómo se imprime un archivo en Linux utilizando CUPS:

1. Un usuario envía un archivo para ser impreso.
2. El demonio de CUPS, `cupsd`, lo envía al *spools* el trabajo de impresión. Este trabajo de impresión recibe un número de trabajo por parte de CUPS, junto con información sobre la cola de impresión que contiene el trabajo, así como el nombre del documento a imprimir.
3. CUPS utiliza *filtros* que están instalados en el sistema para generar un archivo con formato que la impresora puede utilizar.

4. A continuación, CUPS envía el archivo formateado a la impresora para su impresión.

Veremos estos pasos con más detalle, así como la forma de instalar y gestionar una impresora en Linux.

El servicio CUPS

La mayoría de las instalaciones de escritorio de Linux tendrán los paquetes CUPS ya instalados. En las instalaciones mínimas de Linux los paquetes CUPS pueden no estarlo (dependiendo de la distribución). Una instalación básica de CUPS puede realizarse en un sistema Debian con el siguiente:

```
$ sudo apt install cups
```

En los sistemas Fedora el proceso de instalación es igual de sencillo. Tendrá que iniciar el servicio CUPS manualmente después de la instalación en Fedora y otras distribuciones basadas en Red Hat:

```
$ sudo dnf install cups
...
$ sudo systemctl start cups.service
```

Una vez finalizada la instalación, puede comprobar que el servicio CUPS se está ejecutando con el uso del comando `systemctl`:

```
$ systemctl status cups.service
● cups.service - CUPS Scheduler
   Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset:
enabled)
   Active: active (running) since Thu 2020-06-25 14:35:47 EDT; 41min ago
     Docs: man:cupsd(8)
 Main PID: 3136 (cupsd)
    Tasks: 2 (limit: 1119)
   Memory: 3.2M
   CGroup: /system.slice/cups.service
           ├─3136 /usr/sbin/cupsd -l
           └─3175 /usr/lib/cups/notifier/dbus dbus://
```

Como muchos otros demonios de Linux, CUPS depende de un conjunto de archivos de configuración para sus operaciones. A continuación se listan los principales que son de interés para el administrador del sistema:

`/etc/cups/cupsd.conf`

Este archivo contiene los ajustes de configuración para el servicio CUPS. Si está familiarizado con el archivo de configuración del servidor web Apache, el archivo de configuración de CUPS le parecerá bastante similar, ya que utiliza una sintaxis muy parecida. El archivo `cupsd.conf` contiene ajustes para cosas como el control del acceso a las diferentes colas de impresión de uso en el sistema, si la interfaz web de CUPS está o no habilitada, así como el nivel de registro que el demonio utilizará.

`/etc/printcap`

Este es el archivo heredado que fue utilizado por el protocolo LPD (*Line Printer Daemon*) antes de la llegada de CUPS. CUPS todavía creará este archivo en los sistemas para la compatibilidad heredada y es a menudo un enlace simbólico a `/run/cups/printcap`. Cada línea de este archivo contiene una impresora a la que el sistema tiene acceso.

`/etc/cups/printers.conf`

Este archivo contiene cada una de las impresoras configuradas para ser utilizadas por el sistema CUPS. Cada impresora y su cola de impresión asociada en este archivo está encerrada dentro de una sección `<Printer></Printer>`. Este fichero proporciona los listados individuales de impresoras que se encuentran en `/etc/printcap`.

WARNING

No se deben realizar modificaciones en el archivo `/etc/cups/printers.conf` en la línea de comandos mientras el servicio CUPS esté en funcionamiento.

`/etc/cups/ppd/`

No se trata de un archivo de configuración, sino de un directorio que contiene los archivos *PostScript Printer Description* (PPD) de las impresoras que los utilizan. Las capacidades operativas de cada impresora se almacenan en un archivo PPD (que termina con la extensión `.ppd`). Estos archivos son de texto plano y siguen un formato específico.

El servicio CUPS también utiliza el registro de la misma manera que el servicio Apache 2. Los registros se almacenan en `/var/log/cups/` y contienen un `access_log`, `page_log` y un `error_log`. El `access_log` mantiene un registro de los accesos a la interfaz web de CUPS, así como de las acciones realizadas en ella, como la gestión de impresoras. El `page_log` mantiene un registro de los trabajos de impresión que se han enviado a las colas de impresión gestionadas por la instalación de CUPS. El `error_log` contendrá mensajes sobre los trabajos de impresión que han fallado y otros errores registrados por la interfaz web.

A continuación veremos las herramientas y funcionalidades que se utilizan para gestionar el servicio CUPS.

Uso de la interfaz web

Como ya se ha dicho, el archivo de configuración `/etc/cups/cupsd.conf` determina si la interfaz web del sistema CUPS está habilitada. La opción de configuración tiene el siguiente aspecto:

```
# Web interface setting...
WebInterface Yes
```

Si la interfaz web está habilitada, entonces CUPS puede ser gestionado desde un navegador en la URL por defecto de `http://localhost:631`. Por defecto, un usuario del sistema puede ver las impresoras y las colas de impresión, pero cualquier forma de modificación de la configuración requiere un usuario con acceso de root para autenticarse con el servicio web. La sección de configuración dentro del archivo `/etc/cups/cupsd.conf` para restringir el acceso a las capacidades administrativas se parecerá a lo siguiente:

```
# All administration operations require an administrator to authenticate...
<Limit CUPS-Add-Modify-Printer CUPS-Delete-Printer CUPS-Add-Modify-Class CUPS-
Delete-Class CUPS-Set-Default>
  AuthType Default
  Require user @SYSTEM
  Order deny,allow
</Limit>
```

A continuación se desglosan esas opciones:

`AuthType Default`

Utilizará una solicitud de autenticación básica cuando una acción requiera acceso de root.

`Require user @SYSTEM`

Indica que se requerirá un usuario con privilegios administrativos para la operación. Esto podría cambiarse a `@nombredelgrupo` donde los miembros de `nombredelgrupo` pueden administrar el servicio CUPS o se podría proporcionar a los usuarios individuales una lista como en `Require user carol, tim`.

`Order deny,allow`

Se emplea de forma muy parecida a la opción de configuración de Apache 2 donde la acción es denegada por defecto a menos que un usuario (o miembro de un grupo) esté autentificado.

La interfaz web para CUPS se puede desactivar deteniendo primero el servicio CUPS, cambiando la opción `WebInterface` de `Yes` a `No`, y luego reiniciando el servicio CUPS.

La interfaz web de CUPS está construida como un sitio web básico con pestañas de navegación para varias secciones del sistema CUPS. Las pestañas de la interfaz web incluyen lo siguiente:

Home

La página de inicio mostrará la versión actual de CUPS que está instalada. También desglosa CUPS en secciones como:

CUPS for Users

Proporciona una descripción de CUPS, opciones de línea de comandos para trabajar con impresoras y colas de impresión, y un enlace al foro de usuarios de CUPS.

CUPS for Administrators

Proporciona enlaces en la interfaz para instalar y gestionar impresoras y enlaces a información sobre cómo trabajar con impresoras en una red.

CUPS for Developers

Proporciona enlaces para desarrollar el propio CUPS, así como para crear archivos PPD para las impresoras.

Administration

La página de administración también está dividida en secciones:

Printers

Aquí un administrador puede añadir nuevas impresoras al sistema, localizar las impresoras conectadas al sistema y gestionar las que ya están instaladas.

Classes

Las clases son un mecanismo que permite añadir impresoras a grupos con políticas específicas. Por ejemplo, una clase puede contener un grupo de impresoras que pertenecen a una planta específica de un edificio en la que sólo pueden imprimir los usuarios de un departamento concreto. Otra clase puede tener limitaciones en el número de páginas que un usuario puede imprimir. Las clases no se crean por defecto en una instalación de CUPS y tienen que ser definidas por un administrador. Esta es la sección de la interfaz web de CUPS donde se pueden crear y gestionar nuevas clases.

Jobs

Aquí es donde un administrador puede ver todos los trabajos de impresión que están actualmente en cola para todas las impresoras que esta instalación CUPS gestiona.

Server

Aquí es donde un administrador puede hacer cambios en el archivo `/etc/cups/cupsd.conf`. Además, hay otras opciones de configuración disponibles a través de casillas de verificación, como permitir que las impresoras conectadas a esta instalación de CUPS se compartan en una red, la autenticación avanzada y permitir la administración remota de impresoras.

Classes

Si las clases de impresoras están configuradas en el sistema, aparecerán en esta página. Cada clase de impresora tendrá opciones para gestionar todas las impresoras de la clase a la vez, así como para ver todos los trabajos que están en cola para las impresoras de esta clase.

Help

Esta pestaña proporciona enlaces para toda la documentación disponible para CUPS que está instalada en el sistema.

Jobs

La pestaña Trabajos permite buscar trabajos de impresión individuales, así como listar todos los trabajos de impresión actuales gestionados por el servidor.

Printers

La pestaña Impresoras muestra todas las impresoras gestionadas actualmente por el sistema, así como un resumen rápido del estado de cada impresora. Se puede hacer clic en cada una de las impresoras de la lista y el administrador accederá a la página en la que se puede gestionar la impresora en cuestión. La información de las impresoras en esta pestaña proviene del archivo `/etc/cups/printers.conf`.

Instalación de una impresora

Añadir una cola de impresión al sistema es un proceso sencillo dentro de la interfaz web de CUPS:

1. Haga clic en la pestaña **Administración** y luego en el botón **Agregar impresora**.
2. La siguiente página ofrecerá varias opciones dependiendo de cómo esté conectada la impresora a su sistema. Si se trata de una impresora local, seleccione la opción más relevante, como el puerto al que está conectada la impresora o el software de impresión de terceros que pueda estar instalado. CUPS también intentará detectar las impresoras que están conectadas a la red y las mostrará aquí. También puede elegir una opción de conexión directa a una impresora de red en función de los protocolos de impresión en red que admita la impresora. Seleccione la opción adecuada y haga clic en el botón **Continuar**.
3. La siguiente página le permitirá proporcionar un nombre, una descripción y una ubicación

(como “oficina trasera” o “escritorio principal”, etc.) para la impresora. Si desea compartir esta impresora a través de la red, puede seleccionar la casilla de verificación para esa opción en esta página también. Una vez introducida la configuración, haga clic en el botón **Continuar**.

4. En la siguiente página se puede seleccionar la marca y el modelo de la impresora. Esto permite a CUPS buscar en su base de datos instalada localmente los controladores y archivos PPD más adecuados para utilizar con la impresora. Si tiene un archivo PPD proporcionado por el fabricante de la impresora, busque su ubicación y selecciónelo para utilizarlo aquí. Una vez hecho esto, haga clic en el botón **Agregar impresora**.

5. La última página es donde se establecen las opciones por defecto, como el tamaño de página que utilizará la impresora y la resolución de los caracteres impresos en la página. Haga clic en el botón **Establecer opciones por defecto** y su impresora ya está instalada en el sistema.

NOTE Muchas instalaciones de escritorio de Linux tendrán diferentes herramientas que se pueden utilizar para instalar una impresora. Los entornos de escritorio GNOME y KDE tienen sus propias aplicaciones incorporadas que pueden utilizarse para instalar y gestionar impresoras. Además, algunas distribuciones proporcionan aplicaciones de gestión de impresoras por separado. Sin embargo, cuando se trata de una instalación de servidor en la que muchos usuarios van a imprimir, la interfaz web CUPS puede proporcionar las mejores herramientas para la tarea.

La cola de una impresora también puede instalarse utilizando los comandos LPD/LPR heredados. Aquí hay un ejemplo usando el comando `lpadmin`:

```
$ sudo lpadmin -p ENVY-4510 -L "office" -v socket://192.168.150.25 -m everywhere
```

Vamos a desglosar el comando para ilustrar las opciones utilizadas aquí:

- Dado que la adición de una impresora al sistema requiere un usuario con privilegios administrativos, anteponemos al comando `lpadmin` la palabra `sudo`.
- La opción `-p` es el destino de los trabajos de impresión. Es esencialmente un nombre amigable para que el usuario sepa dónde aterrizarán los trabajos de impresión. Típicamente puede proporcionar el nombre de la impresora.
- La opción `-L` es la ubicación de la impresora. Esto es opcional, pero es útil en caso de que tenga que gestionar varias impresoras en diferentes lugares.
- La opción `-v` es para el URI del dispositivo de impresión. El URI del dispositivo es lo que la cola de impresión de CUPS necesita para enviar los trabajos de impresión realizados a una impresora específica. En nuestro ejemplo, estamos utilizando una ubicación de red empleando la dirección

IP proporcionada.

- La última opción, `-m`, se establece como "everywhere". Esto establece el modelo de la impresora para que CUPS determine qué archivo PPD debe utilizar. En las versiones modernas de CUPS, es mejor utilizar "everywhere" para que CUPS pueda comprobar el URI del dispositivo (establecido con la opción anterior `-v`) para determinar automáticamente el archivo PPD correcto a utilizar para la impresora. En situaciones modernas, CUPS simplemente utilizará IPP como se explica a continuación.

Como se ha dicho anteriormente, es mejor dejar que CUPS determine automáticamente qué archivo PPD debe utilizar para una cola de impresión en particular. Sin embargo, el comando (heredado) `lpinfo` puede ser utilizado para consultar los archivos PPD instalados localmente para ver cuáles están disponibles. Simplemente proporcione la opción `--make-and-model` para la impresora que desea instalar y la opción `-m`:

```
$ lpinfo --make-and-model "HP Envy 4510" -m
hplip:0/ppd/hplip/HP/hp-envy_4510_series-hpijs.ppd HP Envy 4510 Series hpijs,
3.17.10
hplip:1/ppd/hplip/HP/hp-envy_4510_series-hpijs.ppd HP Envy 4510 Series hpijs,
3.17.10
hplip:2/ppd/hplip/HP/hp-envy_4510_series-hpijs.ppd HP Envy 4510 Series hpijs,
3.17.10
drv:///hpcups.crv/hp-envy_4510_series.ppd HP Envy 4510 Series, hpcups 3.17.10
everywhere IPP Everywhere
```

Tenga en cuenta que el comando `lpinfo` está obsoleto. Se muestra aquí como un ejemplo de listado de los archivos del controlador de impresión que podría utilizar una impresora.

WARNING	Las futuras versiones de CUPS han dejado de lado los controladores y en su lugar se centrarán en el uso de IPP (*Protocolo de Impresión de Internet*) y los formatos de archivo estándar. La salida del comando anterior ilustra esto con la capacidad de impresión `everywhere IPP Everywhere`. IPP puede realizar las mismas tareas para las que se utiliza un controlador de impresión. IPP, al igual que la interfaz web de CUPS, utiliza el puerto de red 631 con el protocolo TCP.

Se puede establecer una impresora por defecto utilizando el comando `lpoptions`. De esta manera, si la mayoría (o todos) los trabajos de impresión se envían a una impresora en particular, la especificada con el comando `lpoptions` será la predeterminada. Sólo hay que especificar la impresora junto con la opción `-d`:

```
$ lpoptions -d ENVY-4510
```

Gestión de impresoras

Una vez instalada una impresora, el administrador puede utilizar la interfaz web para gestionar las opciones disponibles para la impresora. Un enfoque más directo para la gestión de una impresora es mediante el uso del comando `lpadmin`.

Una opción es permitir que una impresora sea compartida en la red. Esto se puede conseguir con la opción `printer-is-shared`, y especificando la impresora con la opción `-p`:

```
$ sudo lpadmin -p FRONT-DESK -o printer-is-shared=true
```

Un administrador también puede configurar una cola de impresión para que sólo acepte trabajos de impresión de usuarios específicos con cada usuario separado por una coma:

```
$ sudo lpadmin -p FRONT-DESK -u allow:carol,frank,grace
```

A la inversa, sólo se podría denegar el acceso a una cola de impresión específica a determinados usuarios:

```
$ sudo lpadmin -p FRONT-DESK -u deny:dave
```

Los grupos de usuarios también pueden utilizarse para permitir o denegar el acceso a la cola de una impresora siempre que el nombre del grupo se encuentre precedido de una “arroba” (`@`):

```
$ sudo lpadmin -p FRONT-DESK -u deny:@sales,@marketing
```

Una cola de impresión también puede tener una política de error en caso de encontrar problemas para imprimir un trabajo. Con el uso de políticas, un trabajo de impresión puede ser abortado (`abort-job`) o puede haber otro intento de impresión en un momento posterior (`retry-job`). Otras políticas incluyen la capacidad de detener la impresora inmediatamente si se produce un error (`stop-printer`), así como la capacidad de reintentar el trabajo inmediatamente después de detectar un fallo (`retry-current-job`). A continuación se muestra un ejemplo en el que la política de la impresora se establece para abortar el trabajo de impresión si se produce un error en la impresora `FRONT-DESK`:

```
$ sudo lpadmin -p FRONT-DESK -o printer-error-policy=abort-job
```

Asegúrese de revisar las páginas del manual del comando `lpadmin` ubicado en `lpadmin(8)` para obtener más detalles sobre el uso de este.

Envío de trabajos de impresión

Muchas aplicaciones de escritorio le permitirán enviar trabajos de impresión desde un elemento del menú o utilizando el atajo de teclado `Ctrl` + `p`. Si te encuentra en un sistema Linux que no utiliza un entorno de escritorio, todavía puede enviar archivos a una impresora por medio de los comandos LPD/LPR heredados.

El comando `lpr` ("line printer remote") se utiliza para enviar un trabajo de impresión a la cola de una impresora. La forma más básica de utilizar el comando, es colocar el nombre de archivo junto con el comando `lpr`:

```
$ lpr report.txt
```

El comando anterior enviará el archivo `report.txt` a la cola de impresión por defecto del sistema (identificada por el archivo `/etc/cups/printers.conf`).

Si una instalación de CUPS tiene varias impresoras instaladas, se puede utilizar el comando `lpstat` para imprimir una lista de impresoras disponibles utilizando la opción `-p` y la opción `-d` indicará cuál es la impresora por defecto:

```
$ lpstat -p -d
printer FRONT-DESK is idle.  enabled since Mon 03 Aug 2020 10:33:07 AM EDT
printer PostScript_oc0303387803 disabled since Sat 07 Mar 2020 08:33:11 PM EST -
    reason unknown
printer ENVY-4510 is idle.  enabled since Fri 31 Jul 2020 10:08:31 AM EDT
system default destination: ENVY-4510
```

Así, en nuestro ejemplo, el archivo `report.txt` se enviará a la impresora `ENVY-4510`, ya que está configurada por defecto. Si el archivo necesita ser impreso en una impresora diferente, especifique la impresora junto con la opción `P`:

```
$ lpr -P FRONT-DESK report.txt
```

Cuando se envía un trabajo de impresión a CUPS, el demonio averiguará qué backend es el más adecuado para manejar la tarea. CUPS puede hacer uso de varios controladores de impresoras, filtros, monitores de puerto de hardware y otro software para renderizar adecuadamente el documento. Habrá ocasiones en las que un usuario que imprima un documento necesitará hacer modificaciones a *cómo* debe imprimirse el documento. Muchas aplicaciones gráficas facilitan esta tarea. También hay opciones de línea de comandos que pueden ser utilizadas para cambiar la forma en que un documento debe ser impreso. Cuando se envía un trabajo de impresión a través de la línea de comandos, podría utilizar `-o` (de “opciones”) junto con términos específicos para ajustar el diseño del documento para su impresión. A continuación se presenta una breve lista de las opciones más utilizadas:

`landscape`

El documento se imprime con la página girada 90 grados en el sentido de las agujas del reloj. La opción `orientation-requested=4` conseguirá el mismo resultado.

`two-sided-long-edge`

La impresora imprimirá el documento en modo vertical en ambas caras del papel, siempre que la impresora admita esta capacidad.

`two-sided-short-edge`

La impresora imprimirá el documento en modo apaisado en ambas caras del papel, siempre que la impresora admita esta capacidad.

`media`

La impresora imprimirá el trabajo en el tamaño de soporte especificado. Los tamaños de soporte disponibles para un trabajo de impresión dependen de la impresora, pero aquí hay una lista de tamaños comunes:

Opción de tamaño	Propósito
`A4`	ISO A4
`Letter`	US Letter
`Legal`	US Legal
`DL`	ISO DL Envelope
`COM10`	US #10 Envelope

`collate`

Intercalar el documento impreso. Esto es útil si tiene un documento de varias páginas que se imprimirá más de una vez, ya que todas las páginas de cada documento se imprimirán en orden.

Configure esta opción como `true` para activarla o `false` para desactivarla.

`page-ranges`

Esta opción se puede utilizar para seleccionar una sola página a imprimir, o un conjunto específico de páginas a imprimir de un documento. Un ejemplo sería el siguiente `-o page-ranges=5-7,9,15`. Esto imprimiría las páginas 5, 6 y 7 y luego las páginas 9 y 15

`fit-to-page`

Imprima el documento de forma que el archivo se ajuste al papel. Si el archivo que se va a imprimir no proporciona información sobre el tamaño de la página, es posible que el trabajo impreso se escale de forma incorrecta y que partes del documento se salgan de la página o que el documento se escale demasiado.

`outputorder`

Imprime el documento en orden `inverso` o `normal` para comenzar la impresión en la página uno. Si una impresora imprime sus páginas boca abajo, el orden por defecto es `-o outputorder=normal` mientras que las impresoras que imprimen con sus páginas hacia arriba imprimirán con `-o outputorder=reverse`.

Tomando una muestra de las opciones anteriores, se puede construir el siguiente comando de ejemplo:

```
$ lpr -P ACCOUNTING-LASERJET -o landscape -o media=A4 -o two-sided-short-edge
finance-report.pdf
```

Se puede imprimir más de una copia de un documento utilizando la opción de número con el siguiente formato: `-#N` donde `N` es igual al número de copias a imprimir. A continuación se muestra un ejemplo con la opción de intercalar en el que se deben imprimir siete copias de un informe en la impresora por defecto:

```
$ lpr -#7 -o collate=true status-report.pdf
```

Además del comando `lpr`, también se puede utilizar el comando `lp`. Muchas de las opciones que se utilizan con el comando `lpr` también se pueden utilizar con el comando `lp`, pero hay algunas diferencias. Asegúrese de consultar la página de manual en `lp(1)` como referencia. Así es como podemos ejecutar el comando `lpr` de ejemplo anterior utilizando la sintaxis del comando `lp` y especificando también la impresora de destino con la opción `-d`:

```
$ lp -d ACCOUNTING-LASERJET -n 7 -o collate=true status-report.pdf
```

Gestión de los trabajos de impresión

Como se ha dicho anteriormente, cada trabajo de impresión enviado a la cola de impresión recibe un ID de trabajo de CUPS. Un usuario puede ver los trabajos de impresión que ha enviado con el comando `lpq`. Pasando la opción `-a` se mostrarán las colas de todas las impresoras que están gestionadas por la instalación de CUPS:

```
$ lpq -a
Rank        Owner         Job       File(s)                  Total Size
1st         carol         20        finance-report.pdf       5072 bytes
```

El mismo comando `lpstat` utilizado anteriormente también tiene una opción para ver las colas de impresión. La opción `-o` por sí misma mostrará todas las colas de impresión, o se puede especificar una cola de impresión por su nombre:

```
$ lp -o
ACCOUNTING-LASERJET-4                   carol        19456   Wed 05 Aug 2020
04:29:44 PM EDT
```

El ID del trabajo de impresión se le añadirá el nombre de la cola a la que se envió el trabajo, el nombre del usuario que lo envió, el tamaño del archivo y la hora a la que se envió.

Si un trabajo de impresión se atasca en una impresora o un usuario desea cancelar su trabajo de impresión, utilice el comando `lprm` junto con el ID del trabajo encontrado en el comando `lpq`:

```
$ lprm 20
```

Todos los trabajos de una cola de impresión pueden ser eliminados a la vez con sólo un guión `-`:

```
$ lprm -
```

Alternativamente, el comando `cancel` de CUPS también podría ser utilizado por un usuario para detener su trabajo de impresión actual:

```
$ cancel
```

Un trabajo de impresión específico puede ser cancelado por su ID de trabajo precedido por el nombre de la impresora:

```
$ cancel ACCOUNTING-LASERJET-20
```

Un trabajo de impresión también puede moverse de una cola de impresión a otra. Esto suele ser útil en caso de que una impresora deje de responder o el documento a imprimir requiera características disponibles en una impresora diferente. Tenga en cuenta que este procedimiento suele requerir un usuario con privilegios elevados. Utilizando el mismo trabajo de impresión del ejemplo anterior, podríamos moverlo a la cola de la impresora `FRONT-DESK`:

```
$ sudo lpmove ACCOUNTING-LASERJET-20 FRONT-DESK
```

Eliminación de impresoras

Para eliminar una impresora, a menudo es útil listar primero todas las impresoras que están actualmente gestionadas por el servicio CUPS. Esto se puede hacer con el comando `lpstat`:

```
$ lpstat -v
device for FRONT-DESK: socket://192.168.150.24
device for ENVY-4510: socket://192.168.150.25
device for PostScript_oc0303387803: ///dev/null
```

La opción `-v` no sólo muestra las impresoras sino también dónde (y cómo) están conectadas. Es una buena práctica rechazar primero cualquier trabajo nuevo que vaya a la impresora y así proporcionar una razón de por qué la impresora no aceptará nuevos trabajos. Esto se puede hacer con lo siguiente:

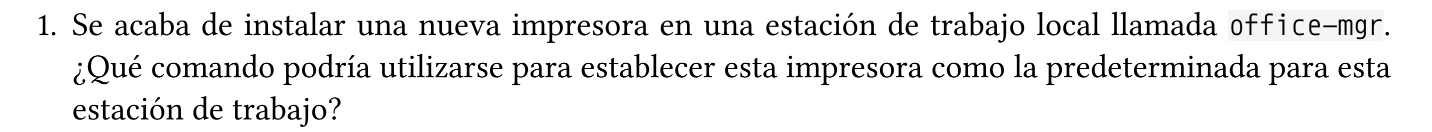
```
$ sudo cupsreject -r "Printer to be removed" FRONT-DESK
```

Tenga en cuenta el uso de `sudo` ya que esta tarea requiere un usuario con privilegios elevados.

Para eliminar una impresora, utilizamos el comando `lpadmin` con la opción `-x` para eliminar la impresora:

```
$ sudo lpadmin -x FRONT-DESK
```

Ejercicios guiados



1. Se acaba de instalar una nueva impresora en una estación de trabajo local llamada `office-mgr`. ¿Qué comando podría utilizarse para establecer esta impresora como la predeterminada para esta estación de trabajo?

2. ¿Qué comando y opción se utilizaría para determinar qué impresoras están disponibles para imprimir desde una estación de trabajo?

3. Utilizando el comando `cancel`, ¿cómo eliminaría un trabajo de impresión con ID 15 que está atascado en la cola de la impresora llamada `office-mgr`?

4. Usted tiene un trabajo de impresión destinado a una impresora que no tiene suficiente papel para imprimir el archivo completo. ¿Qué comando utilizaría para mover el trabajo de impresión con ID 2 en cola para imprimir en la impresora `FRONT-DESK` a la cola de impresión para la impresora `ACCOUNTING-LASERJET`?

Ejercicios de exploración

Utilizando el gestor de paquetes de su distribución, instale los paquetes `cups` y `printer-driver-cups-pdf`. Tenga en cuenta que si está utilizando una distribución basada en Red Hat (como Fedora) el controlador CUPS PDF se llama `cups-pdf`. También instale el paquete `cups-client` para utilizar los comandos de impresión de estilo System V. Utilizaremos estos paquetes para practicar la gestión de una impresora CUPS sin instalar físicamente una impresora real.

1. Compruebe que el demonio CUPS se está ejecutando y, a continuación, verifique que la impresora PDF está activada y configurada por defecto.

2. Ejecute un comando que imprima el archivo `/etc/services`. Ahora debería tener un directorio llamado `PDF` dentro de su directorio principal.

3. Utilice un comando que sólo desactive la impresora y, a continuación, ejecute otro comando que muestre toda la información de estado para verificar que la impresora PDF está desactivada. A continuación, intente imprimir una copia de su archivo `/etc/fstab`. ¿Qué ocurre?

4. Ahora intente imprimir una copia del archivo `/etc/fstab` en la impresora PDF. ¿Qué ocurre?

5. Cancele el trabajo de impresión y, a continuación, elimine la impresora PDF.

Resumen

El demonio CUPS es una plataforma ampliamente utilizada para imprimir en impresoras locales y remotas. Aunque sustituye al protocolo heredado LPD, sigue ofreciendo compatibilidad con versiones anteriores de sus herramientas.

Los archivos y comandos discutidos en esta lección fueron:

`/etc/cups/cupsd.conf`
: El archivo de configuración principal para el servicio CUPS. Este archivo también controla el acceso a la interfaz web de CUPS.

`/etc/printcap`
: Un archivo heredado utilizado por LPD que contiene una línea para cada impresora conectada al sistema.

`/etc/cups/printers.conf`
: El archivo de configuración utilizado por CUPS para la información de la impresora.

La interfaz web de CUPS, que en una instalación por defecto puede encontrarse en `http://localhost:631`. Recuerde que el puerto de red por defecto para la interfaz web es el 631/TCP.

También se trataron los siguientes comandos LPD/LPR heredados:

`lpadmin`
: Se utiliza para instalar y eliminar impresoras y clases de impresoras.

`lpoptions`
: Se utiliza para imprimir las opciones de la impresora y para modificar la configuración de una impresora.

`lpstat`
: Se utiliza para mostrar información de estado de las impresoras conectadas a una instalación CUPS.

`lpr`
: Se utiliza para enviar trabajos de impresión a la cola de una impresora.

`lp`
: Se utiliza para enviar trabajos de impresión (Igual que `lpr`).

`lpq`

Este comando lista los trabajos de impresión dentro de la cola de impresión.

`lprm`

Se utiliza para cancelar trabajos de impresión por ID. El ID de un trabajo se puede obtener con la salida del comando `lpq`.

`cancel`

Una alternativa al comando `lprm` para cancelar trabajos de impresión por su ID.

Asegúrese de revisar las siguientes páginas man para las diferentes herramientas y utilidades de cups: `lpadmin(8)`, `lpoptions(1)`, `lpr(1)`, `lpq(1)`, `lprm(1)`, `cancel(1)`, `lpstat(1)`, `cupsenable(8)` y `cupsaccept(8)`. También se recomienda revisar la documentación de ayuda en línea en `http://localhost:631/help`.

Respuestas a los ejercicios guiados

1. Se acaba de instalar una nueva impresora en una estación de trabajo local llamada `office-mgr`. ¿Qué comando podría utilizarse para establecer esta impresora como la predeterminada para esta estación de trabajo?

```
$ lpoptions -d office-mgr
```

2. ¿Qué comando y opción se utilizaría para determinar qué impresoras están disponibles para imprimir desde una estación de trabajo?

```
$ lpstat -p
```

La opción `-p` muestra todas las impresoras disponibles y si están habilitadas para imprimir.

3. Utilizando el comando `cancel`, ¿cómo eliminarías un trabajo de impresión con ID 15 que está atascado en la cola de la impresora llamada `office-mgr`?

```
$ cancel office-mgr-15
```

4. Usted tiene un trabajo de impresión destinado a una impresora que no tiene suficiente papel para imprimir el archivo completo. ¿Qué comando utilizaría para mover el trabajo de impresión con ID 2 en cola para imprimir en la impresora `FRONT-DESK` a la cola de impresión para la impresora `ACCOUNTING-LASERJET`?

```
$ sudo lpmove FRONT-DESK-2 ACCOUNTING-LASERJET
```

Respuestas a los ejercicios de exploración

Utilizando el gestor de paquetes de su distribución, instale los paquetes `cups` y `printer-driver-cups-pdf`. Tenga en cuenta que si está utilizando una distribución basada en Red Hat (como Fedora) el controlador CUPS PDF se llama `cups-pdf`. También instale el paquete `cups-client` para utilizar los comandos de impresión de estilo System V. Utilizaremos estos paquetes para practicar la gestión de una impresora CUPS sin instalar físicamente una impresora real.

1. Compruebe que el demonio CUPS se está ejecutando y, a continuación, verifique que la impresora PDF está activada y configurada por defecto.

   Un método para comprobar la disponibilidad y el estado de la impresora PDF sería ejecutar el siguiente comando:

   ```
   $ lpstat -p -d

   printer PDF is idle.  enabled since Thu 25 Jun 2020 02:36:07 PM EDTi

   system default destination: PDF
   ```

2. Ejecute un comando que imprima el archivo `/etc/services`. Ahora debería tener un directorio llamado `PDF` dentro de su directorio principal.

   ```
   $ lp -d PDF /etc/services
   ```

   Funcionaría. Ahora tendrá una versión en PDF de este archivo dentro del directorio PDF.

3. Utilice un comando que sólo desactive la impresora y, a continuación, ejecute otro comando que muestre toda la información de estado para verificar que la impresora PDF está desactivada.

   ```
   $ sudo cupsdisable PDF
   ```

   Desactivará la impresora.

   A continuación, ejecute el comando `lpstat -t` para obtener un listado completo del estado de la impresora. Debería ser similar a la siguiente salida:

```
$ scheduler is running

system default destination: PDFi

device for PDF: cups-pdf:/

PDF accepting requests since Wed 05 Aug 2020 04:19:15 PM EDTi

printer PDF disabled since Wed 05 Aug 2020 04:19:15 PM EDT -

Paused
```

4. Ahora intente imprimir una copia del archivo `/etc/fstab` en la impresora PDF. ¿Qué sucede?

   Después de intentar el comando `lp -d PDF /etc/fstab` debería obtener una salida que muestre la información del ID del trabajo. Sin embargo, si comprueba la carpeta PDF en su directorio principal, el nuevo archivo no está allí. A continuación, puede comprobar la cola de impresión con el comando `lpstat -o`, y encontrará su trabajo en la lista.

5. Cancele el trabajo de impresión y, a continuación, elimine la impresora PDF.

   Utilizando la salida del comando anterior `lp`, utilice el comando `cancel` para eliminar el trabajo. Por ejemplo:

   ```
   $ cancel PDF-4
   ```

   A continuación, ejecute el comando `lpstat -o` para verificar que el trabajo ha sido eliminado.

   Elimine la impresora PDF con lo siguiente: `sudo lpadmin -x PDF`. A continuación, verifique que la impresora ha sido eliminada: `lpstat -a`.



Tema 109: Fundamentos de redes



109.1 Fundamentos de los protocolos de Internet

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 109.1

Importancia

4

Áreas de conocimiento clave

- Demostrar conocimientos sobre máscaras de red y notación CIDR.
- Conocer la diferencia entre direcciones IP privadas y públicas con notación decimal punteada.
- Conocer puertos y servicios TCP y UDP comunes (20, 21, 22, 23, 25, 53, 80, 110, 123, 139, 143, 161, 162, 389, 443, 465, 514, 636, 993, 995).
- Conocer las diferencias y características principales de los protocolos UDP, TCP e ICMP.
- Conocer las principales diferencias entre IPv4 e IPv6.
- Conocer las características básicas de IPv6.

Lista parcial de archivos, términos y utilidades

- `/etc/services`
- IPv4, IPv6
- Subnetting
- TCP, UDP, ICMP



109.1 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.1 Fundamentos de los protocolos de Internet
Lección:	1 de 2

Introducción

El TCP/IP (*Protocolo de Control de Transmisión/Protocolo de Internet*) es una pila de protocolos que permite la comunicación entre ordenadores. A pesar de su nombre, la pila consta de varios protocolos como IP, TCP, UDP, ICMP, DNS, SMTP, ARP y otros.

IP (Internet Protocol)

El IP es el protocolo responsable del direccionamiento lógico de un host, que permite el envío de paquetes de un host a otro. Para ello, a cada dispositivo de la red se le asigna una dirección IP única, y es posible asignar más de una dirección al mismo dispositivo.

En la versión 4 del protocolo IP, normalmente llamada IPv4, la dirección está formada por un conjunto de 32 bits separados en 4 grupos de 8 bits, representados en forma decimal. Por ejemplo: Formato binario (4 grupos de 8 bits):: `11000000.10101000.00001010.00010100`

Formato decimal

`192.168.10.20`

En IPv4, los valores de cada octeto pueden ir de 0 a 255, lo que equivale a 11111111 en formato binario.

Clases de direcciones

En teoría, las direcciones IP están separadas por clases, que se definen por el rango del primer octeto, como se muestra en la siguiente tabla:

Clase	Primer Octeto	Rango	Ejemplo
A	1-126	`1.0.0.0` – `126.255.255.255`	`10.25.13.10`
B	128-191	`128.0.0.0` – `191.255.255.255`	`141.150.200.1`
C	192-223	`192.0.0.0` – `223.255.255.255`	`200.178.12.242`

Direcciones IP públicas y privadas

Como se ha mencionado anteriormente, para que se produzca la comunicación, cada dispositivo de la red debe estar asociado con al menos una dirección IP única. Sin embargo, si cada dispositivo conectado a Internet en el mundo tuviera una dirección IP única, no habría suficientes direcciones IP (v4) para todos. Por ello, se definieron las direcciones IP *privadas*.

Las direcciones IP privadas son rangos de direcciones IP que se han reservado para su uso en las redes internas (privadas) de empresas, instituciones, hogares, etc. Dentro de la misma red, el uso de una dirección IP sigue siendo único. Sin embargo, la misma dirección IP privada puede ser utilizada dentro de cualquier red privada.

Así, en Internet tenemos un tráfico de datos que utiliza direcciones IP públicas, que son reconocibles y se enrutan a través de Internet, mientras que dentro de las redes privadas se utilizan estos rangos de direcciones IP reservados. El router se encarga de convertir el tráfico de la red privada a la red pública y viceversa.

Los rangos de direcciones IP privadas, separadas por clases, pueden verse en la siguiente tabla:

Clase	Primer Octeto	Rango	Ejemplo
A	1-126	`1.0.0.0` – `126.255.255.255`	`10.0.0.0` – `10.255.255.255`
B	128-191	`128.0.0.0` – `191.255.255.255`	`172.16.0.0` – `172.31.255.255`

Clase	Primer Octeto	Rango	Ejemplo
C	192-223	`192.0.0.0` – `223.255.255.255`	`192.168.0.0` – `192.168.255.255`

Conversión de formato decimal a binario

Para los sujetos de este tema, es importante saber cómo convertir las direcciones IP entre los formatos binario y decimal.

La conversión del formato decimal al binario se realiza mediante divisiones consecutivas por 2. Como ejemplo, convirtamos el valor 105 mediante los siguientes pasos:

1. Dividiendo el valor 105 entre 2 tenemos:

```
105/2
Quotient = 52
Rest = 1
```

2. Divida el cociente secuencialmente por 2, hasta que el cociente sea igual a 1:

```
52/2
Rest = 0
Quotient = 26
```

```
26/2
Rest = 0
Quotient = 13
```

```
13/2
Rest = 1
Quotient = 6
```

```
6/2
Rest = 0
Quotient = 3
```

```
3/2
Rest = 1
Quotient = 1
```

3. Agrupe el último cociente seguido del resto de todas las divisiones:

```
1101001
```

4. Rellene con "0" a la izquierda hasta completar 8 bits:

```
01101001
```

5. Al final, tenemos que el valor 105 en decimal es igual a 01101001 en binario.

Conversión de formato binario a decimal

En este ejemplo, utilizaremos el valor binario `10110000`.

1. Cada bit está asociado a un valor con una potencia de base dos. Las potencias se inician en 0, y se incrementan de derecha a izquierda. En este ejemplo tendremos:

1	0	1	1	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

2. Cuando el bit es 1, asignamos el valor de la potencia respectiva, cuando el bit es 0 el resultado es 0.

1	0	1	1	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	0	32	16	0	0	0	0

3. Sume todos los valores:

   128 + 32 + 16 = 176

4. Así, 10110000 en binario es igual a 176 en decimal.

Máscara de red

La máscara de red se utiliza junto con la dirección IP para determinar qué parte de la dirección representa la red y cuántos hosts. Tiene el mismo formato que las direcciones IP, es decir, hay 32 bits en 4 grupos de 8. Por ejemplo:

Decimal	Binario	CIDR
`255.0.0.0`	`11111111.00000000.00000000.00000000`	`8`
`255.255.0.0`	`11111111.11111111.00000000.00000000`	`16`
`255.255.255.0`	`11111111.11111111.11111111.00000000`	`24`

Utilizando la máscara `255.255.0.0` como ejemplo, indica que en la IP asociada, los primeros 16 bits (2 primeros decimales) identifican la red/subred y los últimos 16 bits se utilizan para identificar de forma única los hosts dentro de la red.

El CIDR (*Classless Inter-Domain Routing*) mencionado anteriormente está relacionado con una notación de máscara simplificada, que indica el número de bits (1) asociados a la red/subred. Esta notación se utiliza habitualmente para sustituir el formato decimal, por ejemplo `/24` en lugar de `255.255.255.0`.

Es interesante observar que cada clase de IP tiene una máscara estándar, como se indica a continuación:

Clase	Primero octeto	Rango	Máscara por defecto
A	1-126	`1.0.0.0` – `126.255.255.255`	`255.0.0.0 / 8`
B	128-191	`128.0.0.0` – `191.255.255.255`	`255.255.0.0 / 16`
C	192-223	`192.0.0.0` – `223.255.255.255`	`255.255.255.0 / 24`

Sin embargo, este patrón no significa que esta sea la máscara que se utilizará siempre. Es posible utilizar cualquier máscara con cualquier dirección IP, como veremos a continuación.

Aquí hay algunos ejemplos de uso de direcciones IP y máscaras:

```
192.168.8.12 / 255.255.255.0 / 24
```

Rango

`192.168.8.0` - `192.168.8.255`

Dirección de la red

`192.168.8.0`

Dirección de Broadcast

`192.168.8.255`

Hosts

`192.168.8.1` - `192.168.8.254`

En este caso tenemos que los 3 primeros dígitos (primeros 24 bits) de la dirección IP definen la red y el último dígito identifica las direcciones de los hosts, es decir, el rango de esta red va de `192.168.8.0` a `192.168.8.255`.

Ahora tenemos dos conceptos importantes: Cada red/subred tiene 2 direcciones reservadas, la primera dirección del rango se llama *dirección de red*. En este caso `192.168.8.0`, que se utiliza para identificar la propia red/subred. La última dirección del rango se llama *dirección de Broadcast (difusión)*, en este caso `192.168.8.255`. Esta dirección de destino se utiliza para enviar el mismo mensaje (paquete) a todos los hosts IP de esa red/subred. Las direcciones de red y de broadcast no pueden ser utilizadas por los dispositivos de la red. Por lo tanto, la lista de direcciones IP que se pueden configurar efectivamente va desde `192.168.8.1` hasta `192.168.8.254`.

Ahora el ejemplo de la misma dirección IP, pero con una máscara diferente:

```
192.168.8.12 / 255.255.0.0 / 16
```

Rango

`192.168.0.0` - `192.168.255.255`

Dirección de la red

`192.168.0.0`

Dirección de Broadcast

`192.168.255.255`

Hosts

`192.168.0.1` – `192.168.255.254`

Vea cómo esta máscara cambia el rango de direcciones IP que están dentro de la misma red/subred.

La división de las redes por máscaras no se limita a los valores por defecto (8, 16, 24). Podemos crear subdivisiones a nuestro gusto, añadiendo o quitando bits en la identificación de la red, creando las nuevas subredes.

Por ejemplo:

```
11111111.11111111.11111111.00000000 = 255.255.255.0 = 24
```

Si queremos subdividir la red anterior en 2, basta con añadir otro bit a la identificación de la red en la máscara, así:

```
11111111.11111111.11111111.10000000 = 255.255.255.128 = 25
```

Tenemos entonces las siguientes subredes:

```
192.168.8.0   - 192.168.8.127
192.168.8.128 - 192.168.8.255
```

Si aumentamos la subdivisión de la red:

```
11111111.11111111.11111111.11000000 = 255.255.255.192 = 26
```

Tendremos:

```
192.168.8.0   - 192.168.8.63
192.168.8.64  - 192.168.8.127
192.168.8.128 - 192.168.8.191
192.168.8.192 - 192.168.8.255
```

Hay que tener en cuenta que en cada subred tendremos las direcciones reservadas de red (la primera del rango) y de broadcast (la última del rango), por lo que cuanto más se subdivida la red, menos IPs podrán ser utilizadas efectivamente por los hosts.

Identificación de las direcciones de red y de difusión

A través de una Dirección IP y una Máscara, podemos identificar la dirección de red y la dirección de difusión (Broadcast) y así definir el rango de direcciones IP para la red/subred.

La dirección de red se obtiene mediante un "AND lógico" entre la dirección IP y la máscara en sus formatos binarios. Tomemos el ejemplo utilizando la IP `192.168.8.12` y la máscara `255.255.255.192`.

Convirtiendo del formato decimal al binario, como vimos anteriormente, tenemos:

```
11000000.10101000.00001000.00001100 (192.168.8.12)
11111111.11111111.11111111.11000000 (255.255.255.192)
```

Con el "AND lógico", tenemos 1 y 1 = 1, 0 y 0 = 0, 1 y 0 = 0, así que:

```
11000000.10101000.00001000.00001100 (192.168.8.12)
11111111.11111111.11111111.11000000 (255.255.255.192)
11000000.10101000.00001000.00000000
```

Así que la dirección de red para esa subred es `192.168.8.0`.

Ahora para obtener la dirección de difusión debemos utilizar la dirección de red donde todos los bits relacionados con la dirección del host a 1:

```
11000000.10101000.00001000.00000000 (192.168.8.0)
11111111.11111111.11111111.11000000 (255.255.255.192)
11000000.10101000.00001000.00111111
```

La dirección de difusión es entonces `192.168.8.63`.

En conclusión, tenemos:

```
192.168.8.12 / 255.255.255.192 / 26
```

Rango

`192.168.8.0` - `192.168.8.63`

Dirección de la red

`192.168.8.0`

Dirección de Broadcast

`192.168.8.63`

Hosts

`192.168.8.1` – `192.168.8.62`

Ruta por defecto

Como hemos visto hasta ahora, las máquinas que están dentro de la misma red/subred lógica pueden comunicarse directamente a través del protocolo IP. Pero consideremos el siguiente ejemplo:

Red 1

`192.168.10.0/24`

Red 2

`192.168.200.0/24`

En este caso, la máquina `192.168.10.20` no puede enviar directamente un paquete a la `192.168.200.100`, ya que están en redes lógicas diferentes. Para permitir esta comunicación se utiliza un router (o un conjunto de routers). Un router en esta configuración también puede llamarse puerta de enlace, ya que proporciona una puerta de enlace entre dos redes. Este dispositivo tiene acceso a ambas redes ya que está configurado con direcciones IP de ambas redes. Por ejemplo `192.168.10.1` y `192.168.200.1`, y por ello consigue ser el intermediario en esta comunicación.

Para ello, cada host de la red debe tener configurada lo que se denomina *ruta por defecto*. La ruta por defecto indica la IP a la que deben enviarse todos los paquetes cuyo destino sea una dirección IP que no forme parte de la red lógica del host.

En el ejemplo anterior, la ruta por defecto para las máquinas de la red `192.168.10.0/24` será `192.168.10.1`, que es la dirección IP del router/gateway, mientras que la ruta por defecto para las máquinas de la red `192.168.200.0/24` será `192.168.200.1`.

La ruta por defecto también se utiliza para que las máquinas de la red privada (LAN) tengan acceso a Internet (WAN), a través de un router.

Ejercicios guiados

1. Utilizando la dirección IP `172.16.30.230` y la máscara de red `255.255.255.224`, identifique:

La notación CIDR para la máscara de red	
Dirección de la red	
Dirección de Broadcast	
Número de direcciones IP que se pueden utilizar para los hosts en esta subred	

2. ¿Qué configuración se requiere en un host para permitir una comunicación IP con un host en una red lógica diferente?

Ejercicios de exploración

1. ¿Por qué los rangos de direcciones IP que empiezan por `127` y el rango posterior a `224` no están incluidos en las clases de direcciones IP A, B o C?

2. Uno de los campos pertenecientes a un paquete IP que es muy importante es el TTL (*Time To Live*). ¿Cuál es la función de este campo y cómo funciona?

3. Explique la función de NAT y cuándo se utiliza.

Resumen

En esta lección se han tratado los principales conceptos del protocolo IPv4, que es el responsable de permitir la comunicación entre los hosts de una red.

También se estudiaron las principales operaciones que el profesional debe conocer para convertir las direcciones IP en diferentes formatos y para poder analizar y realizar las configuraciones lógicas en redes y subredes.

Se abordaron los siguientes temas:

- Clases de direcciones IP
- DIrecciones IP públicas y privadas
- Cómo convertir direcciones IP de formato decimal a binario, y viceversa
- La máscara de red (netmask)
- Cómo identificar las direcciones de red y de difusión a partir de la dirección IP y la máscara de red
- Ruta por defecto

Respuestas a los ejercicios guiados

1. Utilizando la dirección IP `172.16.30.230` y la máscara de red `255.255.255.224`, identifique:

La notación CIDR para la máscara de red	`27`
Dirección de la red	`172.16.30.224`
Dirección de Broadcast	`172.16.30.255`
Número de direcciones IP que se pueden utilizar para los hosts en esta subred	`30`

2. ¿Qué configuración se requiere en un host para permitir una comunicación IP con un host en una red lógica diferente?

   Ruta por defecto

Respuestas a los ejercicios de exploración

1. ¿Por qué los rangos de direcciones IP que empiezan por `127` y el rango posterior a `224` no están incluidos en las clases de direcciones IP A, B o C?

   El rango que comienza con `127` está reservado para direcciones loopback, utilizadas para pruebas y comunicación interna entre procesos, como la dirección `127.0.0.1`. Además, las direcciones por encima de `224` tampoco se utilizan como direcciones de host, sino para multidifusión y otros fines.

2. Uno de los campos pertenecientes a un paquete IP que es muy importante es el TTL (*Time To Live*). ¿Cuál es la función de este campo y cómo funciona?

   El TTL define el tiempo de vida de un paquete. Esto se implementa a través de un contador en el que el valor inicial definido en el origen se decrementa en cada puerta de enlace/enrutador por el que pasa el paquete, que también se denomina “salto”. Si este contador llega a 0, el paquete se descarta.

3. Explique la función de NAT y cuándo se utiliza.

   La función NAT (*Traducción de direcciones de red*) permite a los hosts de una red interna, que utiliza direcciones IP privadas, tener acceso a Internet como si estuvieran conectados directamente a ella, con la IP pública utilizada en la puerta de enlace.



109.1 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.1 Fundamentos de los protocolos de Internet
Lección:	2 de 2

Introducción

Al principio de este subtema vimos que la pila TCP/IP está compuesta por una serie de protocolos diferentes. Hasta ahora hemos estudiado el protocolo IP, que permite la comunicación entre máquinas mediante direcciones IP, máscaras, rutas, etc.

Para que un host pueda acceder a un servicio disponible en otro host, además del protocolo de direccionamiento IP en la capa de red, será necesario utilizar un protocolo en la capa de transporte como los protocolos TCP y UDP.

Estos protocolos realizan esta comunicación a través de puertos de red. Así que además de definir una dirección IP de origen y de destino, se utilizarán los puertos de origen y de destino para acceder a un servicio.

El puerto se identifica mediante un campo de 16 bits, lo que proporciona un límite de 65535 puertos posibles. Los servicios (destino) utilizan los puertos del 1 al 1023, que se denominan *puertos privilegiados* porque tienen acceso de root al sistema. El origen de la conexión utilizará el rango de puertos de 1024 a 65535, llamados *puertos no privilegiados*, o puertos de socket.

Los puertos utilizados por cada tipo de servicio están estandarizados y controlados por IANA

(*Internet Assigned Numbers Authority*). Esto significa que en cualquier sistema, el puerto 22 es utilizado por el servicio SSH, el puerto 80 por el servicio HTTP y así sucesivamente.

La siguiente tabla contiene los principales servicios y sus respectivos puertos.

Puerto	Servicio
20	FTP (data)
21	FTP (control)
22	SSH (Secure Socket Shell)
23	Telnet (Remote connection without encryption)
25	SMTP (Simple Mail Transfer Protocol), Sending Mails
53	DNS (Domain Name System)
80	HTTP (Hypertext Transfer Protocol)
110	POP3 (Post Office Protocol), Receiving Mails
123	NTP (Network Time Protocol)
139	Netbios
143	IMAP (Internet Message Access Protocol), Accessing Mails
161	SNMP (Simple Network Management Protocol)
162	SNMPTRAP, SNMP Notifications
389	LDAP (Lightweight Directory Access Protocol)
443	HTTPS (Secure HTTP)
465	SMTPS (Secure SMTP)
514	RSH (Remote Shell)
636	LDAPS (Secure LDAP)
993	IMAPS (Secure IMAP)
995	POP3S (Secure POP3)

En un sistema Linux, los puertos de servicio estándar aparecen en el archivo `/etc/services`. La identificación del puerto de destino deseado en una conexión se realiza utilizando el caracter `:` (dos puntos) después de la dirección IPv4. Así, cuando se busca acceder al servicio HTTPS que es atendido por el host IP `200.216.10.15`, el cliente debe enviar la solicitud al destino `200.216.10.15:443`.

Los servicios mencionados anteriormente, y todos los demás, utilizan un protocolo de transporte en función de las características requeridas por el servicio, siendo TCP y UDP los principales.

Protocolo de Control de Transmisión (TCP)

TCP es un protocolo de transporte orientado a la conexión. Esto significa que se establece una conexión entre el cliente a través del puerto de socket, y el servicio a través del puerto estándar de servicio. El protocolo se encarga de garantizar que todos los paquetes se entreguen correctamente, verificando la integridad y el orden de los mismos, incluyendo la retransmisión de paquetes perdidos por errores de red.

Así, la aplicación no necesita implementar este control de flujo de datos, ya que está garantizado por el protocolo TCP.

Protocolo de Datagramas de Usuario (UDP)

UDP establece una conexión entre el cliente y el servicio, pero no controla la transmisión de datos de esa conexión. Es decir, no comprueba si los paquetes se han perdido, o si están fuera de servicio, etc. La aplicación es responsable de implementar los controles que sean necesarios.

Al haber menos control, UDP permite un mejor rendimiento en el flujo de datos, lo que es importante para algunos tipos de servicios.

Protocolo de mensajes de control de Internet (ICMP)

ICMP es un protocolo de la capa de red de la pila TCP/IP y su función principal es analizar y controlar los elementos de la red, haciendo posible, por ejemplo:

- Control del volumen de tráfico
- Detección de destinos inalcanzables
- Redirección de rutas
- Comprobación del estado de los hosts remotos

Es el protocolo utilizado por el comando `ping`, que se estudiará en otro subtema.

IPv6

Hasta ahora hemos estudiado la versión 4 del protocolo IP, es decir, IPv4. Esta ha sido la versión estándar utilizada en todos los entornos de red e Internet. Sin embargo, tiene limitaciones, especialmente en lo que respecta al número de direcciones disponibles, y con una realidad ya actual

de que todos los dispositivos estarán de alguna manera conectados a Internet (véase IoT), es cada vez más común utilizar la versión 6 del protocolo IP, comúnmente escrita como IPv6.

IPv6 trae consigo una serie de cambios, nuevas implementaciones y características, así como una nueva representación de la propia dirección.

Cada dirección IPv6 tiene 128 bits, divididos en 8 grupos de 16 bits, representados por valores hexadecimales.

Por ejemplo:

```
2001:0db8:85a3:08d3:1319:8a2e:0370:7344
```

Abreviaturas

IPv6 define formas de acortar las direcciones en algunas situaciones. Revisemos la siguiente dirección:

```
2001:0db8:85a3:0000:0000:0000:0000:7344
```

La primera posibilidad es reducir las cadenas de `0000` a sólo `0`, lo que resulta:

```
2001:0db8:85a3:0:0:0:0:7344
```

Además, en el caso de las cadenas de grupo con un valor de `0`, se pueden omitir, como se indica a continuación:

```
2001:0db8:85a3::7344
```

Sin embargo, esta última abreviatura sólo puede hacerse una vez en la dirección. Véase el ejemplo:

```
2001:0db8:85a3:0000:0000:1319:0000:7344

2001:0db8:85a3:0:0:1319:0:7344

2001:0db8:85a3::1319:0:7344
```

Tipos de direcciones IPv6

IPv6 clasifica las direcciones en 3 tipos:

Unicast
: Identifica una única interfaz de red. Por defecto, los 64 bits de la izquierda identifican la red, y los 64 bits de la derecha identifican la interfaz.

Multicast
: Identifica un conjunto de interfaces de red. Un paquete enviado a una dirección de multidifusión se enviará a todas las interfaces que pertenezcan a ese grupo. Aunque es similar, no debe confundirse con la difusión, no existe en el protocolo IPv6.

Anycast
: Esto también identifica un conjunto de interfaces en la red, pero el paquete reenviado a una dirección *anycast* será entregado sólo a una dirección de ese conjunto, no a todas.

Diferencias entre IPv4 e IPv6

Además de la dirección, se pueden señalar otras diferencias entre las versiones 4 y 6 del protocolo IP. Estas son algunas de ellas:

- Los puertos de servicio siguen los mismos estándares y protocolos (TCP, UDP), la diferencia está sólo en la representación del conjunto de IP y puertos. En IPv6 la dirección IP debe estar protegida con `[]` (corchetes):

  IPv4
  : `200.216.10.15:443`

  IPv6
  : `[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]:443`

- IPv6 no implementa el cometido de difusión exactamente como existe en IPv4. Sin embargo, se puede conseguir el mismo resultado enviando el paquete a la dirección `ff02::1`, llegando a todos los hosts de la red local. Algo similar a utilizar `224.0.0.1` en IPv4 para la multidifusión como destino.
- A través de la función SLAAC (*Stateless Address Autoconfiguration*), los hosts IPv6 son capaces de autoconfigurarse.
- El campo TTL (*Time to Live*) de IPv4 ha sido sustituido por el “Hop Limit” en la cabecera IPv6.
- Todas las interfaces IPv6 tienen una dirección local, llamada dirección link-local, con el prefijo

`fe80::/10`.

- IPv6 implementa el *Neighbor Discovery Protocol* (NDP), que es similar al ARP utilizado por IPv4, pero con mucha más funcionalidad.

Ejercicios guiados

1. ¿Qué puerto es el predeterminado para el protocolo SMTP?

2. ¿Cuántos puertos diferentes hay disponibles en un sistema?

3. ¿Qué protocolo de transporte garantiza que todos los paquetes se entreguen correctamente, verificando la integridad y el orden de los mismos?

4. ¿Qué tipo de dirección IPv6 se utiliza para enviar un paquete a todas las interfaces que pertenecen a un grupo de hosts?

Ejercicios de exploración

1. Mencione 4 ejemplos de servicios que utilizan el protocolo TCP por defecto.

2. ¿Cuál es el nombre del campo en el paquete de cabecera IPv6 que implementa el mismo recurso de TTL en IPv4?

3. ¿Qué tipo de información es capaz de descubrir el Protocolo de Descubrimiento de Vecinos (NDP)?

Resumen

Esta lección cubre los principales protocolos de transporte y servicios utilizados en la pila TCP/IP.

Otro tema importante fue la versión 6 del Protocolo IP, incluyendo las direcciones IPv6 y las principales diferencias con IPv4.

Se abordaron los siguientes temas:

- La correlación entre los números de puerto y los servicios
- TCP (Protocolo de Control de Transmisión)
- UDP (User Datagram Protocol)
- ICMP (Internet Control Message Protocol)
- La dirección IPv6 y cómo se puede abreviar
- Tipos de direcciones IPv6
- Principales diferencias entre IPv4 e IPv6

Respuestas a los ejercicios guiados

1. ¿Qué puerto es el predeterminado para el protocolo SMTP?

   25

2. ¿Cuántos puertos diferentes hay disponibles en un sistema?

   65535

3. ¿Qué protocolo de transporte garantiza que todos los paquetes se entreguen correctamente, verificando la integridad y el orden de los mismos?

   TCP

4. ¿Qué tipo de dirección IPv6 se utiliza para enviar un paquete a todas las interfaces que pertenecen a un grupo de hosts?

   Multicast

Respuestas a los ejercicios de exploración

1. Mencione 4 ejemplos de servicios que utilizan el protocolo TCP por defecto.

   FTP, SMTP, HTTP, POP3, IMAP, SSH

2. ¿Cuál es el nombre del campo en el paquete de cabecera IPv6 que implementa el mismo recurso de TTL en IPv4?

   Hop Limit

3. ¿Qué tipo de información es capaz de descubrir el Protocolo de Descubrimiento de Vecinos (NDP)?

   NDP es capaz de obtener diversa información de la red, incluyendo otros nodos, direcciones duplicadas, rutas, servidores DNS, pasarelas, etc.



109.2 Configuración de red persistente

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 109.2

Importancia

4

Áreas de conocimiento clave

- Entender la configuración TCP/IP básica de un servidor.
- Configurar redes ethernet y wi-fi usando NetworkManager.
- Conocimientos de systemd-networkd.

Lista parcial de archivos, términos y utilidades

- `/etc/hostname`
- `/etc/hosts`
- `/etc/nsswitch.conf`
- `/etc/resolv.conf`
- `nmcli`
- `hostnamectl`
- `ifup`
- `ifdown`



109.2 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.2 Configuración de red persistente
Lección:	1 de 2

Introducción

En cualquier red TCP/IP, cada nodo debe configurar su adaptador de red para que coincida con los requisitos de la red, de lo contrario no podrán comunicarse entre sí. Por lo tanto, el administrador del sistema debe proporcionar la configuración básica para que el sistema operativo sea capaz de configurar la interfaz de red adecuada, así como de identificarse a sí mismo y las características básicas de la red cada vez que se inicie.

Las configuraciones de red son agnósticas con respecto a los sistemas operativos, pero estos últimos tienen sus propios métodos para almacenar y aplicar estas configuraciones. Los sistemas Linux confían en las configuraciones almacenadas en archivos de texto plano bajo el directorio `/etc` para hacer aparecer la conectividad de red durante el arranque. Merece la pena conocer cómo se utilizan estos archivos para evitar la pérdida de conectividad debido a una mala configuración local.

La interfaz de red

Interfaz de red es el término con el que el sistema operativo se refiere al canal de comunicación configurado para trabajar con el hardware de red conectado al sistema, como un dispositivo Ethernet o Wi-Fi. La excepción a esto es la interfaz *loopback*, que el sistema operativo utiliza cuando necesita

establecer una conexión consigo mismo, pero el propósito principal de una interfaz de red es proporcionar una ruta a través de la cual se pueden enviar datos locales y recibir datos remotos. A menos que la interfaz de red esté correctamente configurada, el sistema operativo no podrá comunicarse con otras máquinas de la red.

En la mayoría de los casos, la configuración correcta de la interfaz se define por defecto o se personaliza durante la instalación del sistema operativo. Sin embargo, a menudo es necesario revisar o incluso modificar estos ajustes cuando la comunicación no funciona correctamente o cuando el comportamiento de la interfaz requiere una personalización.

Hay muchos comandos de Linux para listar qué interfaces de red están presentes en el sistema, pero no todos están disponibles en todas las distribuciones. El comando `ip`, es parte del conjunto básico de herramientas de red incluidas en todas las distribuciones de Linux y puede ser utilizado para listar las interfaces de red. El comando completo para mostrar las interfaces es `ip link show`:

```
$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp3s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
DEFAULT group default qlen 1000
        link/ether 00:16:3e:8d:2b:5b brd ff:ff:ff:ff:ff:ff
```

Si está disponible, también se puede utilizar el comando `nmcli device`:

```
$ nmcli device
DEVICE      TYPE      STATE      CONNECTION
enp3s5      ethernet  connected  Gigabit Powerline Adapter
lo          loopback  unmanaged  --
```

Los comandos mostrados en los ejemplos no modifican ninguna configuración del sistema, por lo que pueden ser ejecutados por un usuario sin privilegios. Ambos comandos listan dos interfaces de red: `lo` (la interfaz loopback) y `enp3s5` (una interfaz ethernet).

Las computadoras de mesa y las portátiles con Linux suelen tener dos o tres interfaces de red predefinidas, una para la interfaz virtual de bucle invertido y las otras asignadas al hardware de red por el sistema. Los servidores y dispositivos de red con Linux, en cambio, pueden tener decenas de interfaces de red, pero los mismos principios se aplican a todos ellos. La abstracción proporcionada por el sistema operativo permite configurar las interfaces de red utilizando los mismos métodos, independientemente del hardware subyacente. Sin embargo, conocer los detalles sobre el hardware

subyacente de una interfaz puede ser útil para entender mejor lo que ocurre cuando la comunicación no funciona como se espera. En un sistema en el que hay muchas interfaces de red disponibles, podría no ser obvio cuál corresponde a wi-fi y al ethernet. Por esta razón, Linux utiliza una convención de nomenclatura de interfaces que ayuda a identificar qué interfaz de red corresponde a cada dispositivo y puerto.

Nombres de interfaces

Las antiguas distribuciones de Linux nombraban las interfaces de red ethernet como `eth0`, `eth1`, etc., numeradas según el orden en que el kernel identifica los dispositivos. Las interfaces inalámbricas se llamaban `wlan0`, `wlan1`, etc. Sin embargo, esta convención de nomenclatura, no aclara qué puerto ethernet específico coincide con la interfaz `eth0`, por ejemplo. Dependiendo de cómo se detectara el hardware, era incluso posible que dos interfaces de red intercambiaran sus nombres después de un reinicio.

Para superar esta ambigüedad, los sistemas Linux más recientes emplean una convención de nomenclatura predecible para las interfaces de red, estableciendo una relación más estrecha entre el nombre de la interfaz y la conexión de hardware subyacente.

En las distribuciones de Linux que utilizan el esquema de nomenclatura systemd, todos los nombres de interfaz comienzan con un prefijo de dos caracteres que significa el tipo de interfaz:

`en`
: Ethernet

`ib`
: InfiniBand

`sl`
: Serial line IP (slip)

`wl`
: Wireless local area network (WLAN)

`ww`
: Wireless wide area network (WWAN)

De mayor a menor prioridad, el sistema operativo utiliza las siguientes reglas para nombrar y numerar las interfaces de red:

1. Nombra la interfaz según el índice proporcionado por la BIOS o por el firmware de los

dispositivos integrados, por ejemplo, `eno1`.

2. Designa la interfaz según el índice de la ranura PCI express, tal y como lo indica la BIOS o el firmware, por ejemplo `ens1`.
3. Nombra la interfaz según su dirección en el bus correspondiente, por ejemplo `enp3s5`.
4. Designa la interfaz con la dirección MAC de la misma, por ejemplo `enx78e7d1ea46da`.
5. Nombra la interfaz utilizando la convención heredada, por ejemplo `eth0`.

Es correcto suponer, que la interfaz de red `enp3s5` se denominó así porque no se ajustaba a los dos primeros métodos de denominación, por lo que se utilizó en su lugar su dirección en el bus y la ranura correspondientes. La dirección del dispositivo `03:05.0`, encontrada en la salida del comando `lspci`, revela el dispositivo asociado:

```
$ lspci | fgrep Ethernet
03:05.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8110SC/8169SC
Gigabit Ethernet (rev 10)
```

Las interfaces de red son creadas por el propio kernel de Linux, pero hay muchos comandos que se pueden utilizar para interactuar con ellas. Normalmente, la configuración se realiza de forma automática y no es necesario cambiar la configuración manualmente. Sin embargo, con el nombre de la interfaz, es posible indicarle al kernel cómo proceder para configurarla si es necesario.

Gestión de interfaces

A lo largo de los años, se han desarrollado varios programas para interactuar con las características de red proporcionadas por el núcleo de Linux. Aunque el antiguo comando `ifconfig` todavía se puede utilizar para realizar configuraciones y consultas simples de las interfaces, ahora está obsoleto debido a su limitado soporte de las interfaces que no son Ethernet. El comando `ifconfig` fue sustituido por el comando `ip`, que es capaz de gestionar muchos otros aspectos de las interfaces TCP/IP, como rutas y túneles.

Las muchas capacidades del comando `ip` pueden ser excesivas para la mayoría de las tareas ordinarias, por lo que existen comandos auxiliares para facilitar la activación y configuración de las interfaces de red. Los comandos `ifup` y `ifdown` pueden utilizarse para configurar las interfaces de red basándose en las definiciones de las interfaces que se encuentran en el fichero `/etc/network/interfaces`. Aunque pueden ser invocados manualmente, estos comandos se ejecutan normalmente de forma automática durante el arranque del sistema.

Todas las interfaces de red gestionadas por `ifup` y `ifdown` deben estar listadas en el fichero `/etc/network/interfaces`. El formato utilizado en el fichero es sencillo: las líneas que comienzan

con la palabra `auto` se utilizan para identificar las interfaces físicas que se van a activar cuando se ejecute `ifup` con la opción `-a`. El nombre de la interfaz debe seguir a la palabra `auto` en la misma línea. Todas las interfaces marcadas como `auto` se activan en el momento del arranque, en el orden en que aparecen en la lista.

WARNING

Los métodos de configuración de red utilizados por `ifup` y `ifdown` no están estandarizados en todas las distribuciones de Linux. CentOS, por ejemplo, mantiene la configuración de las interfaces en archivos individuales en el directorio `/etc/sysconfig/network-scripts/` y el formato de configuración utilizado en ellos es ligeramente diferente del formato utilizado en `/etc/network/interfaces`.

La configuración real de la interfaz se escribe en otra línea, empezando por la palabra `iface`, seguida del nombre de la interfaz, el nombre de la familia de direcciones que utiliza la interfaz y el nombre del método utilizado para configurar la interfaz. El siguiente ejemplo muestra un fichero de configuración básico para las interfaces `lo` (loopback) y `enp3s5`:

```
auto lo
iface lo inet loopback

auto enp3s5
iface enp3s5 inet dhcp
```

La familia de direcciones debe ser `inet` para redes TCP/IP, pero también hay soporte para redes IPX (`ipx`), y redes IPv6 (`inet6`). Las interfaces Loopback utilizan el método de configuración `loopback`. Con el método `dhcp`, la interfaz utilizará la configuración IP proporcionada por el servidor DHCP de la red. Los ajustes de la configuración de ejemplo permiten la ejecución del comando `ifup` utilizando el nombre de la interfaz `enp3s5` como argumento:

```
# ifup enp3s5
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp3s5/00:16:3e:8d:2b:5b
Sending on   LPF/enp3s5/00:16:3e:8d:2b:5b
Sending on   Socket/fallback
DHCPDISCOVER on enp3s5 to 255.255.255.255 port 67 interval 4
DHCPOFFER of 10.90.170.158 from 10.90.170.1
DHCPREQUEST for 10.90.170.158 on enp3s5 to 255.255.255.255 port 67
DHCPACK of 10.90.170.158 from 10.90.170.1
bound to 10.90.170.158 -- renewal in 1616 seconds.
```

En este ejemplo, el método elegido para la interfaz `enp3s5` fue `dhcp`, por lo que el comando `ifup` llamó a un programa cliente DHCP para obtener la configuración IP del servidor DHCP. Del mismo modo, el comando `ifdown enp3s5` se puede utilizar para apagar la interfaz.

En redes sin servidor DHCP, se puede utilizar el método `static` en su lugar y proporcionar la configuración IP manualmente en `/etc/network/interfaces`. Por ejemplo:

```
iface enp3s5 inet static
    address 192.168.1.2/24
    gateway 192.168.1.1
```

Las interfaces que utilizan el método `static` no necesitan una directiva `auto` correspondiente, ya que se activan siempre que se detecta el hardware de la red.

Si la misma interfaz tiene más de una entrada `iface`, entonces todas las direcciones y opciones configuradas se aplicarán al abrir esa interfaz. Esto es útil para configurar tanto direcciones IPv4 como IPv6 en la misma interfaz, así como para configurar múltiples direcciones del mismo tipo en una sola interfaz.

Nombres locales y remotos

Una configuración TCP/IP que funcione es sólo el primer paso hacia la plena usabilidad de la red. Además de poder identificar los nodos de la red por sus números IP, el sistema debe ser capaz de identificarlos con nombres más fáciles de entender por los seres humanos.

El nombre con el que se identifica el sistema es personalizable y es una buena práctica definirlo,

incluso si la máquina no está destinada a unirse a una red. El nombre local suele coincidir con el nombre de red de la máquina, pero esto no es necesariamente cierto siempre. Si el fichero `/etc/hostname` existe, el sistema operativo utilizará el contenido de la primera línea como nombre local, que a partir de entonces se llamará simplemente *hostname*. Las líneas que comienzan con `#` dentro de `/etc/hostname` son ignoradas.

El fichero `/etc/hostname` puede editarse directamente, pero el nombre de la máquina también puede definirse con el comando `hostnamectl`. Cuando se suministra con el subcomando `set-hostname`, el comando `hostnamectl` tomará el nombre dado como argumento y lo escribirá en `/etc/hostname`:

```
# hostnamectl set-hostname storage
# cat /etc/hostname
storage
```

El nombre de host definido en `/etc/hostname` es el nombre de host *estático*, es decir, el nombre que se utiliza para inicializar el nombre de host del sistema en el arranque. El nombre de host estático puede ser una cadena de forma libre de hasta 64 caracteres. Sin embargo, se recomienda que conste sólo de caracteres ASCII en minúsculas y sin espacios ni puntos. También debe limitarse al formato permitido para las etiquetas de nombres de dominio DNS, aunque esto no es un requisito estricto.

El comando `hostnamectl` puede establecer otros dos tipos de nombres de host además del nombre de host estático:

Pretty hostname

A diferencia del nombre de host estático, este otro nombre puede incluir todo tipo de caracteres especiales. Se puede utilizar para establecer un nombre más descriptivo para el equipo, por ejemplo, “LAN Shared Storage”:

```
# hostnamectl --pretty set-hostname "LAN Shared Storage"
```

Transient hostname

Se utiliza cuando el nombre de host estático no está establecido o cuando es el nombre `localhost` por defecto. El nombre de host transitorio es normalmente el nombre establecido junto con otras configuraciones automáticas, pero también puede ser modificado por el comando `hostnamectl`, por ejemplo

```
# hostnamectl --transient set-hostname generic-host
```

Si no se utiliza la opción `--pretty` ni `--transient`, los tres tipos de nombres de host se establecerán con el nombre dado. Para establecer el nombre de host estático, pero no los "pretty" y "transient", se debe utilizar la opción `--static`. En todos los casos, sólo el nombre de host estático se almacena en el fichero `/etc/hostname`. El comando `hostnamectl` también se puede utilizar para mostrar varios bits de información descriptiva y de identidad sobre el sistema en ejecución:

```
$ hostnamectl status
     Static hostname: storage
     Pretty hostname: LAN Shared Storage
  Transient hostname: generic-host
           Icon name: computer-server
             Chassis: server
          Machine ID: d91962a957f749bbaf16da3c9c86e093
             Boot ID: 8c11dcab9c3d4f5aa53f4f4e8fdc6318
    Operating System: Debian GNU/Linux 10 (buster)
              Kernel: Linux 4.19.0-8-amd64
        Architecture: x86-64
```

Esta es la acción por defecto del comando `hostnamectl`, por lo que el subcomando `status` puede ser omitido. En cuanto al nombre de los nodos de la red remota, hay dos formas básicas que el sistema operativo puede implementar para hacer coincidir nombres y números IP: utilizar una fuente local o utilizar un servidor remoto para traducir los nombres en números IP y viceversa. Los métodos pueden ser complementarios entre sí y su orden de prioridad se define en el archivo de configuración *Name Service Switch*: `/etc/nsswitch.conf`. Este fichero es utilizado por el sistema y las aplicaciones para determinar no sólo las fuentes de coincidencias nombre-IP, sino también las fuentes de las que puede obtener información de servicios de nombres en una serie de categorías, llamadas *bases de datos*.

La base de datos *hosts* lleva la cuenta del mapeo entre nombres de host y direcciones IPs. La línea dentro de `/etc/nsswitch.conf` que comienza con `hosts` define los servicios responsables de proporcionar las asociaciones para ello:

```
hosts: files dns
```

En esta entrada de ejemplo, `files` y `dns` son los nombres de servicio que especifican cómo funcionará el proceso de búsqueda de nombres de host. En primer lugar, el sistema buscará coincidencias en los archivos locales, y luego preguntará al servicio DNS por las coincidencias.

El archivo local para la base de datos de hosts es `/etc/hosts`, un simple archivo de texto que asocia direcciones IP con nombres de host, una línea por dirección IP, por ejemplo:

```
127.0.0.1 localhost
```

El número de IP *127.0.0.1* es la dirección por defecto de la interfaz loopback, de ahí su asociación con el nombre *localhost*.

También es posible vincular alias opcionales a la misma IP. Los alias pueden proporcionar ortografías alternativas, nombres de host más cortos y deben añadirse al final de la línea, por ejemplo:

```
192.168.1.10 foo.mydomain.org foo
```

Las reglas de formato para el archivo `/etc/hosts` son:

- Los campos de la entrada están separados por cualquier número de espacios en blanco y/o caracteres de tabulación.
- El texto desde un carácter `#` hasta el final de la línea es un comentario y se ignora.
- Los nombres de host sólo pueden contener caracteres alfanuméricos, signos menos y puntos.
- Los nombres de host deben comenzar con un carácter alfabético y terminar con un carácter alfanumérico.

Las direcciones IPv6 también pueden añadirse a `/etc/hosts`. La siguiente entrada se refiere a la dirección IPv6 loopback:

```
::1 localhost ip6-localhost ip6-loopback
```

Tras la especificación del servicio `files`, la especificación `dns` indica al sistema que solicite a un servicio DNS la asociación nombre/IP deseada. El conjunto de rutinas responsables de este método se llama *resolver* y su fichero de configuración es `/etc/resolv.conf`. El siguiente ejemplo muestra un `/etc/resolv.conf` genérico que contiene entradas para los servidores DNS públicos de Google:

```
nameserver 8.8.4.4
nameserver 8.8.8.8
```

Como se muestra en el ejemplo, la palabra clave `nameserver` indica la dirección IP del servidor DNS. Sólo se requiere un servidor de nombres, pero se pueden indicar hasta tres servidores de nombres. Los complementarios se utilizarán como reserva. Si no hay entradas de servidor de nombres, el

comportamiento por defecto es utilizar el servidor de nombres de la máquina local.

El resolvedor puede configurarse para añadir automáticamente el dominio a los nombres antes de consultarlos en el servidor de nombres. Por ejemplo:

```
nameserver 8.8.4.4
nameserver 8.8.8.8
domain mydomain.org
search mydomain.net mydomain.com
```

La entrada `dominio` establece `midominio.org` como nombre de dominio local, por lo que las consultas de nombres dentro de este dominio podrán utilizar nombres cortos relativos al dominio local. La entrada `search` tiene un propósito similar, pero acepta una lista de dominios para probar cuando se proporciona un nombre corto. Por defecto, sólo contiene el nombre del dominio local.

Ejercicios guiados

1. ¿Qué comandos se pueden utilizar para enumerar los adaptadores de red presentes en el sistema?

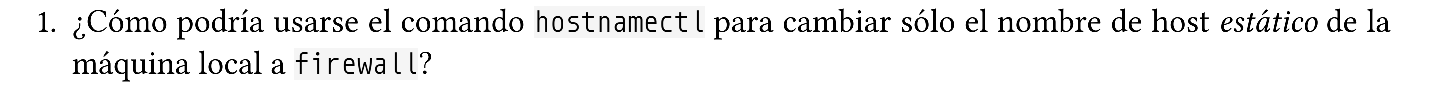
2. ¿Cuál es el tipo de adaptador de red cuyo nombre de interfaz es `wlo1`?

3. ¿Qué papel juega el archivo `/etc/network/interfaces` durante el arranque?

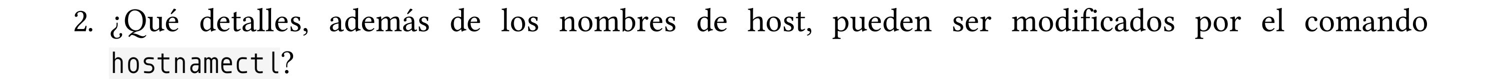
4. ¿Qué entrada en `/etc/network/interfaces` configura la interfaz `eno1` para obtener su configuración IP con DHCP?

Ejercicios de exploración

1. ¿Cómo podría usarse el comando `hostnamectl` para cambiar sólo el nombre de host *estático* de la máquina local a `firewall`?



2. ¿Qué detalles, además de los nombres de host, pueden ser modificados por el comando `hostnamectl`?



3. ¿Qué entrada en `/etc/hosts` asocia los nombres `firewall` y `router` con la IP `10.8.0.1`?

4. ¿Cómo se podría modificar el archivo `/etc/resolv.conf` para enviar todas las peticiones DNS a `1.1.1.1`?

Resumen

Esta lección cubre cómo hacer cambios persistentes en la configuración de la red local usando archivos y comandos estándar de Linux. Linux espera que las configuraciones TCP/IP estén en lugares específicos y puede ser necesario cambiarlas cuando las configuraciones por defecto no son apropiadas. La lección pasa por los siguientes temas:

- Cómo identifica Linux las interfaces de red.
- La activación de interfaces durante el arranque y la configuración IP básica.
- Cómo el sistema operativo asocia los nombres con los hosts.

Los conceptos, comandos y procedimientos abordados fueron:

- Convenciones de nomenclatura de interfaces.
- Listado de interfaces de red con `ip` y `nmcli`.
- Activación de interfaces con `ifup` y `ifdown`.
- El comando `hostnamectl` y el archivo `/etc/hostname`.
- Archivos `/etc/nsswitch.conf`, `/etc/hosts` y `/etc/resolv.conf`.

Respuestas a los ejercicios guiados

1. ¿Qué comandos se pueden utilizar para enumerar los adaptadores de red presentes en el sistema?

   Comandos `ip link show`, `nmcli device` y el legado `ifconfig`.

2. ¿Cuál es el tipo de adaptador de red cuyo nombre de interfaz es `wlo1`?

   El nombre comienza con `wl`, por lo que es un adaptador de LAN inalámbrica.

3. ¿Qué papel juega el archivo `/etc/network/interfaces` durante el arranque?

   Tiene las configuraciones utilizadas por el comando `ifup` para activar las interfaces correspondientes durante el arranque.

4. ¿Qué entrada en `/etc/network/interfaces` configura la interfaz `eno1` para obtener su configuración IP con DHCP?

   La línea `iface eno1 inet dhcp`.

Respuestas a los ejercicios de exploración

1. ¿Cómo podría usarse el comando `hostnamectl` para cambiar sólo el nombre de host *estático* de la máquina local a `firewall`?

   Con la opción `--static`: `hostnamectl --static set-hostname firewall`.

2. ¿Qué detalles, además de los nombres de host, pueden ser modificados por el comando `hostnamectl`?

   `hostnamectl` también puede establecer el icono por defecto de la máquina local, su tipo de chasis, la ubicación y el entorno de despliegue.

3. ¿Qué entrada en `/etc/hosts` asocia los nombres `firewall` y `router` con la IP `10.8.0.1`?

   La línea `10.8.0.1 firewall router`.

4. ¿Cómo se podría modificar el archivo `/etc/resolv.conf` para enviar todas las peticiones DNS a `1.1.1.1`?

   Utilizando `nameserver 1.1.1.1` como su única entrada de servidor de nombres.



109.2 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.2 Configuración de red persistente
Lección:	2 de 2

Introducción

Linux es compatible con prácticamente todas las tecnologías de red utilizadas para conectar servidores, contenedores, máquinas virtuales, ordenadores de sobremesa y dispositivos móviles. Las conexiones entre todos estos nodos de red pueden ser dinámicas y heterogéneas, por lo que requieren una gestión adecuada por parte del sistema operativo que se ejecuta en ellos.

En el pasado, las distribuciones desarrollaban sus propias soluciones personalizadas para gestionar la infraestructura de red dinámica. Hoy en día, herramientas como *NetworkManager* y *systemd* ofrecen funciones más completas e integradas para satisfacer todas las demandas específicas. === NetworkManager

La mayoría de las distribuciones de Linux adoptan el demonio de servicio *NetworkManager* para configurar y controlar las conexiones de red del sistema. El propósito de NetworkManager es hacer que la configuración de la red sea lo más sencilla y automática posible. Cuando se utiliza DHCP, por ejemplo, NetworkManager organiza los cambios de ruta, la obtención de direcciones IP y las actualizaciones de la lista local de servidores DNS, si es necesario. Cuando se dispone de conexiones por cable e inalámbricas. NetworkManager da prioridad por defecto a la conexión por cable e intentará mantener al menos una conexión activa todo el tiempo, siempre que sea posible.

NOTE

Una solicitud mediante DHCP (*Protocolo de configuración dinámica de host*) suele enviarse a través del adaptador de red en cuanto se establece el enlace con la red. El servidor DHCP que está activo en la red responde entonces con la configuración (dirección IP, máscara de red, ruta por defecto, etc.) que el solicitante debe utilizar para comunicarse mediante el protocolo IP.

Por defecto, el demonio NetworkManager controla las interfaces de red no mencionadas en el fichero `/etc/network/interfaces`. Lo hace para no interferir con otros métodos de configuración que puedan estar presentes también, modificando así sólo las interfaces desatendidas.

El servicio NetworkManager se ejecuta en segundo plano con privilegios de root y desencadena las acciones necesarias para mantener el sistema en línea. Los usuarios normales pueden crear y modificar las conexiones de red con aplicaciones cliente que, aunque no tengan privilegios de root, son capaces de comunicarse con el servicio subyacente para realizar las acciones solicitadas.

Las aplicaciones cliente para NetworkManager están disponibles tanto para la línea de comandos como para el entorno gráfico. Para este último, la aplicación cliente viene como un accesorio del entorno de escritorio (bajo nombres como, *nm-tray*, *network-manager-gnome*, *nm-applet* o *plasma-nm*) y suele ser accesible a través de un icono indicador en la esquina de la barra del escritorio o desde la utilidad de configuración del sistema.

En la línea de comandos, el propio NetworkManager proporciona dos programas cliente: `nmcli` y `nmtui`. Ambos programas tienen las mismas características básicas, pero `nmtui` tiene una interfaz basada en curses mientras que `nmcli` es un comando más completo que también puede ser utilizado en scripts. El comando `nmcli` separa todas las propiedades relacionadas con la red controladas por NetworkManager en categorías llamadas *objects*:

`general`

El estado y las operaciones generales de NetworkManager.

`networking`

Control general de la red.

`radio`

Conmutadores de radio NetworkManager.

`connection`

Las conexiones del NetworkManager.

`device`

Dispositivos gestionados por NetworkManager.

`agent`

Agente secreto NetworkManager o agente polkit.

`monitor`

Supervisar los cambios del NetworkManager.

El nombre del objeto es el argumento principal del comando `nmcli`. Para mostrar el estado general de conectividad del sistema, por ejemplo, se debe dar como argumento el objeto `general`:

```
$ nmcli general
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          enabled  enabled  enabled  enabled
```

La columna `STATE` indica si el sistema está conectado a una red o no. Si la conexión está limitada debido a una mala configuración externa o a restricciones de acceso, la columna `CONNECTIVITY` no informará del estado de conectividad `completa`. Si aparece `Portal` en la columna `CONNECTIVITY`, significa que se requieren pasos adicionales de autenticación (normalmente a través del navegador web) para completar el proceso de conexión. Las columnas restantes informan del estado de las conexiones inalámbricas (si las hay), ya sean `WIFI` o `WAN` (Wide Wireless Area Network, es decir, redes celulares). El sufijo `HW` indica que el estado corresponde al dispositivo de red y no a la conexión de red del sistema, es decir, indica si el hardware está activado o desactivado para ahorrar energía.

`nmcli` también necesita un argumento de comando para ejecutarse. El comando `status` se utiliza por defecto si no hay ningún argumento de comando, por lo que el comando `nmcli general` se interpreta en realidad como `nmcli general status`.

No es necesario realizar ninguna acción cuando el adaptador de red se conecta directamente al punto de acceso a través de cables, pero las redes inalámbricas requieren una mayor interacción para aceptar nuevos miembros. `nmcli` facilita el proceso de conexión y guarda la configuración para conectarse automáticamente en el futuro, por lo que es muy útil para los ordenadores portátiles o cualquier otro aparato móvil.

Antes de conectarse al wi-fi, es conveniente listar primero las redes disponibles en el área local. Si el sistema tiene un adaptador wi-fi en funcionamiento, entonces el objeto `device` lo utilizará para escanear las redes disponibles con el comando `nmcli device wifi list`:

```
$ nmcli device wifi list
IN-USE  BSSID              SSID        MODE   CHAN  RATE        SIGNAL  BARS
SECURITY
        90:F6:52:C5:FA:12  Hypnotoad   Infra  11    130 Mbit/s  67      ▯▯▯_   WPA2
        10:72:23:C7:27:AC  Jumbao      Infra  1     130 Mbit/s  55      ▯▯__   WPA2
        00:1F:33:33:E9:BE  NETGEAR     Infra  1     54 Mbit/s   35      ▯▯__   WPA1
WPA2
        A4:33:D7:85:6D:B0  AP53        Infra  11    130 Mbit/s  32      ▯▯__   WPA1
WPA2
        98:1E:19:1D:CC:3A  Bruma       Infra  1     195 Mbit/s  22      ▯___   WPA1
WPA2
```

La mayoría de los usuarios probablemente utilizarán el nombre de la columna `SSID` para identificar la red de interés. Por ejemplo, el comando `nmcli` puede conectarse a la red llamada `Hypnotoad` utilizando de nuevo el objeto `device`:

```
$ nmcli device wifi connect Hypnotoad
```

Si el comando se ejecuta dentro de un emulador de terminal en el entorno gráfico, aparecerá un cuadro de diálogo solicitando la frase de acceso a la red. Cuando se ejecuta en una consola de sólo texto, la contraseña puede ser proporcionada junto con los otros argumentos:

```
$ nmcli device wifi connect Hypnotoad password MyPassword
```

Si la red wi-fi oculta su nombre SSID, `nmcli` aún puede conectarse a ella con los argumentos extra `hidden yes`:

```
$ nmcli device wifi connect Hypnotoad password MyPassword hidden yes
```

Si el sistema tiene más de un adaptador wi-fi, se puede indicar el que se va a utilizar con `ifname`. Por ejemplo, para conectarse usando el adaptador llamado `wlo1`:

```
$ nmcli device wifi connect Hypnotoad password MyPassword ifname wlo1
```

Después de que la conexión tenga éxito, NetworkManager le dará el nombre del SSID correspondiente (si es una conexión wi-fi) y lo conservará para futuras conexiones. Los nombres de las conexiones y sus UUIDs son listados por el comando `nmcli connection show`:

```
$ nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Ethernet            53440255-567e-300d-9922-b28f0786f56e  ethernet  enp3s5
tun0                cae685e1-b0c4-405a-8ece-6d424e1fb5f8  tun       tun0
Hypnotoad           6fdec048-bcc5-490a-832b-da83d8cb7915  wifi      wlo1
4G                  a2cf4460-0cb7-42e3-8df3-ccb927f2fd88  gsm       --
```

Se muestra el tipo de cada conexión — que puede ser `ethernet`, `wifi`, `tun`, `gsm`, `bridge`, etc. — así como el dispositivo al que están asociadas. Para realizar acciones sobre una conexión concreta, hay que proporcionar su nombre o UUID. Para desactivar la conexión `Hypnotoad`, por ejemplo:

```
$ nmcli connection down Hypnotoad
Connection 'Hypnotoad' successfully deactivated
```

Igualmente, el comando `nmcli connection up Hypnotoad` puede ser utilizado para traer la conexión, ya que ahora está guardada por NetworkManager. El nombre de la interfaz también se puede utilizar para reconectar, pero en este caso se debe utilizar el objeto `device` en su lugar:

```
$ nmcli device disconnect wlo2
Device 'wlo1' successfully disconnected.
```

El nombre de la interfaz también puede utilizarse para restablecer la conexión:

```
$ nmcli device connect wlo2
Device 'wlo1' successfully activated with '833692de-377e-4f91-a3dc-d9a2b1fcf6cb'.
```

Tenga en cuenta que el UUID de la conexión cambia cada vez que se abre la conexión, por lo que es preferible utilizar su nombre para mantener la coherencia.

Si el adaptador inalámbrico está disponible pero no se está utilizando, entonces se puede apagar para ahorrar energía. Esta vez, el objeto *radio* debe ser pasado a `nmcli`:

```
$ nmcli radio wifi off
```

Por supuesto, el dispositivo inalámbrico se puede volver a encender con el comando `nmcli radio wifi on`.

Una vez establecidas las conexiones no será necesaria ninguna interacción manual en el futuro, ya que NetworkManager identifica las redes conocidas disponibles y se conecta automáticamente a ellas. Si es necesario, NetworkManager tiene plugins que pueden ampliar sus funcionalidades, como el plugin para soportar conexiones VPN.

systemd-networkd

Los sistemas que ejecutan systemd pueden utilizar opcionalmente sus demonios incorporados para gestionar la conectividad de red: `systemd-networkd` para controlar las interfaces de red y `systemd-resolved` para gestionar la resolución de nombres locales. Estos servicios son compatibles con los métodos de configuración heredados de Linux, pero la configuración de las interfaces de red en particular tiene características que vale la pena conocer.

Los archivos de configuración utilizados por systemd-networkd para configurar las interfaces de red pueden encontrarse en cualquiera de los tres directorios siguientes:

`/lib/systemd/network`
: El directorio de la red del sistema.

`/run/systemd/network`
: El directorio de red volátil en tiempo de ejecución.

`/etc/systemd/network`
: El directorio de red de la administración local.

Los archivos se procesan en orden lexicográfico, por lo que se recomienda comenzar sus nombres con números para facilitar la lectura y el ordenamiento.

Los archivos en `/etc` tienen la mayor prioridad, mientras que los archivos en `/run` tienen prioridad sobre los archivos con el mismo nombre en `/lib`. Esto significa que si los archivos de configuración en diferentes directorios tienen el mismo nombre, entonces systemd-networkd ignorará los archivos con menor prioridad. Separar los archivos de esta manera es una forma de cambiar la configuración de la interfaz sin tener que modificar los archivos originales: se pueden colocar modificaciones en `/etc/systemd/network` para anular las de `/lib/systemd/network`.

El propósito de cada archivo de configuración depende de su sufijo. Los archivos que terminan en `.netdev` son utilizados por systemd-networkd para crear dispositivos de red virtuales, como los dispositivos *bridge* o *tun*. Los archivos que terminan en `.link` establecen configuraciones de bajo nivel para la interfaz de red correspondiente. systemd-networkd detecta y configura los dispositivos de red automáticamente a medida que aparecen — además de ignorar los dispositivos ya configurados por otros medios — por lo que no es necesario añadir estos archivos en la mayoría de las situaciones.

El sufijo más importante es `.network`. Los archivos que utilizan este sufijo pueden utilizarse para configurar direcciones y rutas de red. Al igual que con los otros tipos de archivos de configuración, el nombre del archivo define el orden en el que se procesará el archivo. La interfaz de red a la que se refiere el fichero de configuración se define en la sección `[Match]` dentro del mismo.

Por ejemplo, la interfaz de red ethernet `enp3s5` puede ser seleccionada dentro del archivo `/etc/systemd/network/30-lan.network` utilizando la entrada `Name=enp3s5` en la sección `[Match]`:

```
[Match]
Name=enp3s5
```

También se acepta una lista de nombres separados por espacios en blanco para hacer coincidir muchas interfaces de red con este mismo archivo a la vez. Los nombres pueden contener globos de estilo shell, como `es*`. Otras entradas proporcionan varias reglas de coincidencia, como la selección de un dispositivo de red por su dirección MAC:

```
[Match]
MACAddress=00:16:3e:8d:2b:5b
```

La configuración del dispositivo se encuentra en la sección `[Network]` del archivo. Una modificación de red estática simple sólo requiere las entradas `Address` y `Gateway`:

```
[Match]
MACAddress=00:16:3e:8d:2b:5b

[Network]
Address=192.168.0.100/24
Gateway=192.168.0.1
```

Para utilizar el protocolo DHCP en lugar de direcciones IP estáticas, se debe utilizar la entrada `DHCP`:

```
[Match]
MACAddress=00:16:3e:8d:2b:5b

[Network]
DHCP=yes
```

El servicio systemd-networkd intentará obtener tanto direcciones IPv4 como IPv6 para la interfaz de red. Para utilizar sólo IPv4, se debe utilizar `DHCP=ipv4`. Del mismo modo, `DHCP=ipv6` ignorará la

configuración de IPv4 y utilizará únicamente la dirección IPv6 proporcionada.

Las redes inalámbricas protegidas por contraseña también pueden ser configuradas por systemd-networkd, pero el adaptador de red debe estar ya autenticado en la red antes de que systemd-networkd pueda configurarlo. La autenticación la realiza *WPA supplicant*, un programa dedicado a configurar adaptadores de red para redes protegidas por contraseña.

El primer paso es crear el archivo de credenciales con el comando `wpa_passphrase`:

```
# wpa_passphrase MyWifi > /etc/wpa_supplicant/wpa_supplicant-wlo1.conf
```

Este comando tomará la frase de contraseña para la red inalámbrica `MyWifi` de la entrada estándar y almacenará su hash en el archivo `/etc/wpa_supplicant/wpa_supplicant-wlo1.conf`. Tenga en cuenta que el nombre del archivo debe contener el nombre apropiado de la interfaz inalámbrica, de ahí el `wlo1` en el nombre del archivo.

El gestor systemd lee los archivos de frases de paso WPA en `/etc/wpa_supplicant/` y crea el servicio correspondiente para ejecutar WPA supplicant y poner en marcha la interfaz. El archivo de frases de paso creado en el ejemplo tendrá entonces una unidad de servicio correspondiente llamada `wpa_supplicant@wlo1.service`. El comando `systemctl start wpa_supplicant@wlo1.service` asociará el adaptador inalámbrico con el punto de acceso remoto. El comando `systemctl enable wpa_supplicant@wlo1.service` hace que la asociación sea automática durante el arranque. Finalmente, un archivo `.network` que coincida con la interfaz `wlo1` debe estar presente en `/etc/systemd/network/`, ya que systemd-networkd lo utilizará para configurar la interfaz tan pronto como el suplicante WPA finalice la asociación con el punto de acceso.

Ejercicios guiados

1. ¿Qué significa la palabra `Portal` en la columna `CONNECTIVITY` en la salida del comando `nmcli general status`?



2. En un terminal de consola, ¿cómo puede un usuario normal utilizar el comando `nmcli` para conectarse a la red inalámbrica `MyWifi` protegida por la contraseña `MyPassword`?

3. ¿Qué comando puede encender el adaptador inalámbrico si el sistema operativo lo ha desactivado previamente?

4. ¿En qué directorio deben colocarse los archivos de configuración personalizados cuando systemd-networkd gestiona las interfaces de red?

Ejercicios de exploración

1. ¿Cómo puede un usuario ejecutar el comando `nmcli` para eliminar una conexión no utilizada llamada `Hotel Internet`?

2. NetworkManager escanea las redes wi-fi periódicamente y el comando `nmcli device wifi list` sólo lista los puntos de acceso encontrados en el último escaneo. ¿Cómo debería usarse el comando `nmcli` para pedir a NetworkManager que vuelva a escanear inmediatamente todos los puntos de acceso disponibles?

3. ¿Qué entrada `name` debe utilizarse en la sección `[Match]` de un archivo de configuración systemd-networkd para que coincida con todas las interfaces ethernet?

4. ¿Cómo debe ejecutarse el comando `wpa_passphrase` para utilizar la frase de paso dada como argumento y no desde la entrada estándar?

Resumen

Esta lección cubre las herramientas comunes utilizadas en Linux para gestionar conexiones de red heterogéneas y dinámicas. Aunque la mayoría de los métodos de configuración no requieren la intervención del usuario, a veces es necesaria y herramientas como *NetworkManager* y *systemd-networkd* pueden reducir las molestias al mínimo. La lección repasa los siguientes temas: * Como se integran NetworkManager y systemd-networkd en el sistema. * Como el usuario puede interactuar con NetworkManager y systemd-networkd. * Configuración básica de la interfaz tanto con NetworkManager como con systemd-networkd.

Los conceptos, comandos y procedimientos abordados fueron:

- Comandos de cliente de NetworkManager: `nmtui` y `nmcli`.
- Escaneo y conexión a redes inalámbricas usando los comandos apropiados de `nmcli`.
- Conexiones persistentes a redes wi-fi usando systemd-networkd.

Respuestas a los ejercicios guiados

1. ¿Qué significa la palabra `Portal` en la columna `CONNECTIVITY` en la salida del comando `nmcli general status`?

   Significa que se requieren pasos adicionales de autenticación (normalmente a través del navegador web) para completar el proceso de conexión.

2. En un terminal de consola, ¿cómo puede un usuario normal utilizar el comando `nmcli` para conectarse a la red inalámbrica `MyWifi` protegida por la contraseña `MyPassword`?

   En un terminal de texto, el comando sería

   ```
   $ nmcli device wifi connect MyWifi password MyPassword
   ```

3. ¿Qué comando puede encender el adaptador inalámbrico si el sistema operativo lo ha desactivado previamente?

   ```
   $ nmcli radio wifi on
   ```

4. ¿En qué directorio deben colocarse los archivos de configuración personalizados cuando systemd-networkd gestiona las interfaces de red?

   En el directorio de red de la administración local: `/etc/systemd/network`.

Respuestas a los ejercicios de exploración

1. ¿Cómo puede un usuario ejecutar el comando `nmcli` para eliminar una conexión no utilizada llamada `Hotel Internet`?

   ```
   $ nmcli connection delete "Hotel Internet"
   ```

2. NetworkManager escanea las redes wi-fi periódicamente y el comando `nmcli device wifi list` sólo lista los puntos de acceso encontrados en el último escaneo. ¿Cómo debería usarse el comando `nmcli` para pedir a NetworkManager que vuelva a escanear inmediatamente todos los puntos de acceso disponibles?

   El usuario root puede ejecutar `nmcli device wifi rescan` para que NetworkManager vuelva a escanear los puntos de acceso disponibles.

3. ¿Qué entrada `name` debe utilizarse en la sección `[Match]` de un archivo de configuración systemd-networkd para que coincida con todas las interfaces ethernet?

   La entrada `name=en*`, ya que `en` es el prefijo de las interfaces ethernet en Linux y systemd-networkd acepta globos de tipo shell.

4. ¿Cómo debe ejecutarse el comando `wpa_passphrase` para utilizar la frase de paso dada como argumento y no desde la entrada estándar?

   La contraseña debe darse justo después del SSID, como en `wpa_passphrase MyWifi MyPassword`.



109.3 Resolución de problemas básicos de red

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 109.3

Importancia

4

Áreas de conocimiento clave

- Configurar interfaces de red de forma manual, incluyendo la visualización y modificación de las mismas usando iproute2.
- Configurar tablas de enrutamiento de forma manual, incluyendo la visualización y modificación de las mismas así como la configuración de la ruta predeterminada usando iproute2.
- Depurar problemas relacionados con la configuración de red.
- Conocimientos de los comandos heredados de net-tools.

Lista parcial de archivos, términos y utilidades

- `ip`
- `hostname`
- `ss`
- `ping`
- `ping6`
- `traceroute`
- `traceroute6`

* `tracepath`
* `tracepath6`
* `netcat`
* `ifconfig`
* `netstat`
* `route`



109.3 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.3 Resolución de problemas básicos de red
Lección:	1 de 2

Introducción

Linux tiene unas capacidades de red muy flexibles y potentes. De hecho, los sistemas operativos basados en Linux se utilizan a menudo en los dispositivos de red habituales, incluidos los equipos comerciales más caros. La creación de redes con Linux podría ser una certificación en sí misma. Teniendo esto en cuenta, esta lección sólo va a cubrir algunas herramientas básicas de configuración y resolución de problemas.

Asegúrase de revisar las lecciones sobre protocolos de Internet y configuración de redes persistentes antes de continuar. A continuación, cubriremos las herramientas para configurar y solucionar problemas de redes IPv4 e IPv6.

Aunque no es un objetivo oficial, los *packet sniffers* como `tcpdump` son herramientas útiles para la resolución de problemas. Los rastreadores de paquetes permiten ver y grabar los datos que entran o salen de una interfaz de red. Herramientas como *hex viewers* y *protocol analyzers* pueden ser usadas para ver estos paquetes con más detalle de lo que un sniffer de paquetes suele permitir. No estaría de más conocer estos programas.

Sobre el comando `ip`

El comando `ip` es una utilidad bastante reciente que se utiliza para ver y configurar casi todo lo relacionado con las configuraciones de red. Esta lección cubre algunos de los subcomandos más usados de `ip`, pero apenas es una mirada de lo que está disponible. Aprender a leer la documentación le ayudará a ser mucho más eficiente con ella.

Cada subcomando de `ip` tiene su propia página de manual. La sección `SEE ALSO` de la página man de `ip` tiene una lista de ellos:

```
$ man ip
...
SEE ALSO
       ip-address(8), ip-addrlabel(8), ip-l2tp(8), ip-link(8), ip-maddress(8),
       ip-monitor(8), ip-mroute(8), ip-neighbour(8), ip-netns(8), ip-
       ntable(8), ip-route(8), ip-rule(8), ip-tcp_metrics(8), ip-token(8), ip-
       tunnel(8), ip-xfrm(8)
       IP Command reference ip-cref.ps
...
```

En lugar de mirar esto cada vez que necesite la página man, simplemente añada `-` y el nombre del subcomando a `ip`, por ejemplo, `man ip-route`.

Otra fuente de información es la función de ayuda. Para ver la ayuda integrada, añada `help` después del subcomando:

```
$ ip address help
Usage: ip address {add|change|replace} IFADDR dev IFNAME [ LIFETIME ]
                                                      [ CONFFLAG-LIST ]
       ip address del IFADDR dev IFNAME [mngtmpaddr]
       ip address {save|flush} [ dev IFNAME ] [ scope SCOPE-ID ]
                            [ to PREFIX ] [ FLAG-LIST ] [ label LABEL ] [up]
       ip address [ show [ dev IFNAME ] [ scope SCOPE-ID ] [ master DEVICE ]
                         [ type TYPE ] [ to PREFIX ] [ FLAG-LIST ]
                         [ label LABEL ] [up] [ vrf NAME ] ]
       ip address {showdump|restore}
IFADDR := PREFIX | ADDR peer PREFIX
...
```

Revisión de máscaras de red y enrutamiento

IPv4 e IPv6 son lo que se conoce como protocolos enrutados. Esto significa que están diseñados de tal manera que los diseñadores de redes pueden controlar el flujo de tráfico. Ethernet no es un protocolo enrutable. Esto significa que si se conecta un grupo de dispositivos usando sólo Ethernet, hay muy poco que se pueda hacer para controlar el flujo de tráfico de la red. Cualquier medida para controlar el tráfico acabaría siendo similar a la de los protocolos enrutables y de enrutamiento actuales.

Los protocolos enrutables permiten a los diseñadores de redes segmentarlas para reducir los requisitos de procesamiento de los dispositivos de conectividad, proporcionar redundancia y gestionar el tráfico.

Las direcciones IPv4 e IPv6 tienen dos secciones. El primer conjunto de bits constituye la sección de red, mientras que el segundo conjunto constituye la parte de host. El número de bits que componen la parte de red viene determinado por la *máscara de red* (también llamada *máscara de subred*). A veces también se denomina *longitud del prefijo*. Independientemente de cómo se llame, es el número de bits que la máquina trata como la parte de red de la dirección. Con IPv4, a veces se especifica en notación decimal con puntos.

A continuación se muestra un ejemplo utilizando IPv4. Observe cómo los dígitos binarios mantienen su valor de posición en los octetos incluso cuando se divide por la máscara de red.

```
192.168.130.5/20

          192      168      130      5
          11000000 10101000 10000010 00000101

20 bits = 11111111 11111111 11110000 00000000

Network = 192.168.128.0
Host    = 2.5
```

La parte de red de una dirección es utilizada por las máquinas IPv4 o IPv6 para buscar en su tabla de enrutamiento la interfaz por la que debe enviarse un paquete. Cuando un host IPv4 o IPv6 con el enrutamiento activado recibe un paquete que no es para el propio host, intenta hacer coincidir la parte de red del destino con una red en la tabla de enrutamiento. Si se ubica una entrada que coincida, envía el paquete al destino especificado en la tabla de enrutamiento. Si no se encuentra ninguna entrada y se ha configurado una ruta por defecto, se envía a la ruta por defecto. Si no se localiza ninguna entrada y no se ha configurado ninguna ruta por defecto, el paquete se descarta.

Configurar una interfaz

Hay dos herramientas que cubriremos que puede usar para configurar una interfaz de red: `ifconfig` o `ip`. El programa `ifconfig`, aunque sigue siendo ampliamente utilizado, se considera una herramienta heredada y puede no estar disponible en los sistemas más nuevos.

TIP En las nuevas distribuciones de Linux, la instalación del paquete `net-tools` le proporcionará los comandos de red heredados.

Antes de configurar una interfaz, debe saber qué interfaces están disponibles. Hay varias formas de hacerlo. Una forma es utilizar la opción `-a` de `ifconfig`:

```
$ ifconfig -a
```

Otra forma es con `ip`. A veces verá ejemplos con `ip addr`, `ip a`, y algunos con `ip address`, son sinónimos. Oficialmente, el subcomando es `ip address`. Esto significa que si desea ver la página del manual, debe utilizar `man ip-address` y no `man ip-addr`.

El subcomando `link` para `ip` listará los enlaces de interfaz disponibles para su configuración:

```
$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT group default qlen 1000
    link/ether 08:00:27:54:18:57 brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT group default qlen 1000
    link/ether 08:00:27:ab:11:3e brd ff:ff:ff:ff:ff:ff
```

Asumiendo que el sistema de archivos `sys` está montado, también puede listar el contenido de `/sys/class/net`:

```
$ ls /sys/class/net
enp0s3  enp0s8  lo
```

Para configurar una interfaz con `ifconfig`, debe iniciar la sesión como root o utilizar una herramienta como `sudo` para ejecutar el comando con privilegios de root. Siga el siguiente ejemplo:

```
# ifconfig enp1s0 192.168.50.50/24
```

La versión de Linux de `ifconfig` es flexible con la forma de especificar la máscara de subred:

```
# ifconfig eth2 192.168.50.50 netmask 255.255.255.0
# ifconfig eth2 192.168.50.50 netmask 0xffffff00
# ifconfig enp0s8 add 2001:db8::10/64
```

Revise que con IPv6 usted ha utilizado la palabra clave `add`. Si no precede una dirección IPv6 con `add`, recibirá un mensaje de error.

El siguiente comando configura una interfaz con `ip`:

```
# ip addr add 192.168.5.5/24 dev enp0s8
# ip addr add 2001:db8::10/64 dev enp0s8
```

Con `ip`, se utiliza el mismo comando tanto para IPv4 como para IPv6.

Configuración de opciones de bajo nivel

El comando `ip link` se utiliza para configurar la interfaz de bajo nivel o los ajustes de protocolo como VLANs, ARP, o MTUs, o deshabilitar una interfaz.

Una tarea común para `ip link` es desactivar o activar una interfaz. Esto también se puede hacer con `ifconfig`:

```
# ip link set dev enp0s8 down
# ip link show dev enp0s8
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT
group default qlen 1000
    link/ether 08:00:27:ab:11:3e brd ff:ff:ff:ff:ff:ff
# ifconfig enp0s8 up
# ip link show dev enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT group default qlen 1000
    link/ether 08:00:27:ab:11:3e brd ff:ff:ff:ff:ff:ff
```

A veces puede ser necesario ajustar la MTU de una interfaz. Al igual que con la habilitación/deshabilitación de interfaces, esto puede hacerse con `ifconfig` o `ip link`:

```
# ip link set enp0s8 mtu 2000
# ip link show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2000 qdisc pfifo_fast state UP
mode DEFAULT group default qlen 1000
    link/ether 08:00:27:54:53:59 brd ff:ff:ff:ff:ff:ff
# ifconfig enp0s3 mtu 1500
# ip link show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT group default qlen 1000
    link/ether 08:00:27:54:53:59 brd ff:ff:ff:ff:ff:ff
```

La tabla de enrutamiento

Los comandos `route`, `netstat -r`, o `ip route` pueden ser utilizados para ver su tabla de rutas. Si desea modificar sus rutas, debe utilizar `route` o `ip route`. A continuación se muestran ejemplos de visualización de una tabla de enrutamiento:

```
$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         10.0.2.2        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
192.168.150.0   0.0.0.0         255.255.255.0   U         0 0          0 enp0s8
$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
192.168.150.0/24 dev enp0s8 proto kernel scope link src 192.168.150.200
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
192.168.150.0   0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
```

Fíjese en que no hay ninguna salida relativa a IPv6. Si desea ver su tabla de enrutamiento para IPv6, debe utilizar `route -6`, `netstat -6r`, o `ip -6 route`.


```
$ route -6
Kernel IPv6 routing table
Destination                    Next Hop                   Flag Met Ref Use If
2001:db8::/64                  [::]                       U    256 0     0 enp0s8
fe80::/64                      [::]                       U    100 0     0 enp0s3
2002:a00::/24                  [::]                       !n   1024 0     0 lo
[::]/0                         2001:db8::1                UG   1   0     0 enp0s8
localhost/128                  [::]                       Un   0   2    84 lo
2001:db8::10/128               [::]                       Un   0   1     0 lo
fe80::a00:27ff:fe54:5359/128   [::]                       Un   0   1     0 lo
ff00::/8                       [::]                       U    256 1     3 enp0s3
ff00::/8                       [::]                       U    256 1     6 enp0s8
```

Se ha omitido un ejemplo de `netstat -r6` porque su salida es idéntica a `route -6`. Parte de la salida del comando `route` anterior se explica por sí misma. La columna `Flag` proporciona alguna información sobre la ruta. La columna flags, `U` indica que la ruta está activa. `!` significa que la ruta ha sido rechazada, es decir, una ruta con una bandera `!` no será utilizada. `n` significa que la ruta no ha sido cacheada. El kernel mantiene una caché de rutas para búsquedas más rápidas por separado de todas las rutas conocidas. `G` indica una puerta de enlace. La columna `Metric` o `Met` no es utilizada por el kernel. Se refiere a la distancia administrativa al objetivo. Esta distancia administrativa es utilizada por los protocolos de enrutamiento para determinar las rutas dinámicas. La columna `Ref` es el recuento de referencias, o el número de usos de una ruta. Al igual que `Metric`, no es utilizada por el kernel de Linux. La columna `Use` muestra el número de búsquedas de una ruta.

En la salida de `netstat -r`, `MSS` indica el tamaño máximo de segmento para las conexiones TCP sobre esa ruta. La columna `Window` muestra el tamaño predeterminado de la ventana TCP. La columna `irtt` muestra el tiempo de ida y vuelta de los paquetes en esta ruta.

La salida de `ip route` o `ip -6 route` es la siguiente:

1. Destino.
2. Dirección opcional seguida de interfaz.
3. El protocolo de enrutamiento utilizado para añadir la ruta.
4. El ámbito de la ruta. Si se omite, se trata de un ámbito global o de una puerta de enlace.
5. La métrica de la ruta. Esta es utilizada por los protocolos de enrutamiento dinámico para determinar el coste de la ruta. La mayoría de los sistemas no la utilizan.
6. Si es una ruta IPv6, la preferencia de ruta RFC4191.

Algunos ejemplos lo aclararán:

Ejemplo de IPv4

```
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
```

1. El destino es la ruta por defecto.
2. La dirección de la puerta de enlace es `10.0.2.2` alcanzable a través de la interfaz `enp0s3`.
3. Ha sido añadida a la tabla de enrutamiento por DHCP.
4. Se ha omitido el ámbito, por lo que es global.
5. La ruta tiene un valor de coste de `100`.
6. No hay preferencia de ruta IPv6.

Ejemplo de IPv6

```
fc0::/64 dev enp0s8 proto kernel metric 256 pref medium
```

1. El destino es `fc0::/64`.
2. Es alcanzable a través de la interfaz `enp0s8`.
3. Ha sido añadida automáticamente por el kernel.
4. Se ha omitido el ámbito, por lo que es global.
5. La ruta tiene un valor de coste de `256`.
6. Tiene una preferencia IPv6 de `media`.

Gestión de rutas

Las rutas pueden ser gestionadas utilizando `route` o `ip route`. A continuación se muestra un ejemplo de cómo añadir y eliminar una ruta utilizando el comando `route`. Con `route`, debe utilizar la opción `-6` para IPv6:

```
# ping6 -c 2 2001:db8:1::20
connect: Network is unreachable
# route -6 add 2001:db8:1::/64 gw 2001:db8::3
# ping6 -c 2 2001:db8:1::20
PING 2001:db8:1::20(2001:db8:1::20) 56 data bytes
64 bytes from 2001:db8:1::20: icmp_seq=1 ttl=64 time=0.451 ms
64 bytes from 2001:db8:1::20: icmp_seq=2 ttl=64 time=0.438 ms
# route -6 del 2001:db8:1::/64 gw 2001:db8::3
# ping6 -c 2 2001:db8:1::20
connect: Network is unreachable
```

A continuación se muestra el mismo ejemplo utilizando el comando `ip route`:

```
# ping6 -c 2 2001:db8:1:20
connect: Network is unreachable
# ip route add 2001:db8:1::/64 via 2001:db8::3
# ping6 -c 2 2001:db8:1:20
PING 2001:db8:1::20(2001:db8:1::20) 56 data bytes
64 bytes from 2001:db8:1::20: icmp_seq=2 ttl=64 time=0.529 ms
64 bytes from 2001:db8:1::20: icmp_seq=2 ttl=64 time=0.438 ms
# ip route del 2001:db8:1::/64 via 2001:db8::3
# ping6 -c 2 2001:db8:1::20
connect: Network is unreachable
```

Ejercicios guiados

1. ¿Qué comandos se pueden utilizar para listar las interfaces de red?

2. ¿Cómo se desactiva temporalmente una interfaz? ¿Cómo se vuelve a habilitar?

3. ¿Cuál de las siguientes es una máscara de subred razonable para IPv4?

`0.0.0.255`	
`255.0.255.0`	
`255.252.0.0`	
`/24`	

4. ¿Qué comandos puede utilizar para verificar su ruta por defecto?

5. ¿Cómo se añade una segunda dirección IP a una interfaz?

Ejercicios de exploración

1. ¿Qué subcomando de `ip` se puede utilizar para configurar el etiquetado vlan?

2. ¿Cómo se configura una ruta por defecto?

3. ¿Cómo se puede obtener información detallada sobre el comando `ip neighbour`? ¿Qué sucede si lo ejecuta por sí mismo?

4. ¿Cómo se hace una copia de seguridad de la tabla de enrutamiento? ¿Cómo se restaura desde ella?

5. ¿Qué subcomando `ip` puede utilizarse para configurar las opciones del árbol de expansión?

Resumen

La red se configura normalmente mediante los scripts de inicio del sistema o un ayudante como NetworkManager. La mayoría de las distribuciones tienen herramientas que editarán los archivos de configuración de los scripts de inicio por usted. Consulte la documentación de su distribución para más detalles.

Ser capaz de configurar manualmente la red le permite solucionar los problemas de forma más eficaz. Es útil en entornos mínimos utilizados para cosas como la restauración de copias de seguridad o la migración a un nuevo hardware.

Las utilidades que se cubren en esta sección tienen más funcionalidades que las ya anteriormente mencionadas en la lección. Valdría la pena hojear la página de manual de cada una para familiarizarse con las opciones disponibles. Los comandos `ss` e `ip` son herramientas modernas y recomendadas, mientras que el resto de las que se cubren, aunque todavía son de uso común, se consideran herramientas heredadas.

La mejor manera de familiarizarse con las herramientas cubiertas es la práctica. Utilizando un ordenador con una cantidad modesta de RAM, es posible configurar un laboratorio de red virtual utilizando máquinas virtuales con las que se puede practicar. Tres máquinas virtuales son suficientes para familiarizarse con las herramientas enumeradas.

Los comandos utilizados en esta lección incluyen:

`ifconfig`
: Utilidad heredada para configurar las interfaces de red y revisar sus estados.

`ip`
: Moderna y versátil utilidad para configurar las interfaces de red y revisar sus estados.

`netstat`
: Comando heredado utilizado para ver las conexiones de red actuales y la información de las rutas.

`route`
: Comando heredado utilizado para ver o modificar la tabla de enrutamiento de un sistema.

Respuestas a los ejercicios guiados

1. ¿Qué comandos se pueden utilizar para listar las interfaces de red?

   Cualquiera de los siguientes comandos:

   `ip link`, `ifconfig -a`, o `ls /sys/class/net`

2. ¿Cómo se desactiva temporalmente una interfaz? ¿Cómo se vuelve a habilitar?

   Puede utilizar `ifconfig` o `ip link`:

   Utilizando `ifconfig`:

   ```
   $ ifconfig wlan1 down
   $ ifconfig wlan1 up
   ```

   Utilizando `ip link`:

   ```
   $ ip link set wlan1 down
   $ ip link set wlan1 up
   ```

3. ¿Cuál de las siguientes es una máscara de subred razonable para IPv4?

   - `255.252.0.0`
   - `/24`

     Las otras máscaras de las listas no son válidas porque no separan la dirección limpiamente en dos secciones, la primera parte define la red y la segunda el host. Los bits más a la izquierda de una máscara siempre serán `1` y los bits de la derecha siempre serán `0`.

4. ¿Qué comandos puede utilizar para verificar su ruta por defecto?

   Puede utilizar `route`, `netstat -r` o `ip route`:

```
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         server          0.0.0.0         UG    600    0        0 wlan1
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlan1
$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         server          0.0.0.0         UG        0 0          0 wlan1
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 wlan1
$ ip route
default via 192.168.1.20 dev wlan1 proto static metric 600
192.168.1.0/24 dev wlan1 proto kernel scope link src 192.168.1.24 metric 600
```

5. ¿Cómo se añade una segunda dirección IP a una interfaz?

   Se usaría `ip address` o `ifconfig`. Tomando en cuenta que `ifconfig` es una herramienta heredada:

```
$ ip addr add 172.16.15.16/16 dev enp0s9 label enp0s9:sub1
```

   La parte del comando `label enp0s9:sub1` añade un alias a `enp0s9`. Si no usa la herramienta heredada `ifconfig` puede omitir esto. Si lo hace, el comando seguirá funcionando, pero la dirección que acaba de añadir no aparecerá en la salida de `ifconfig`.

   También puede utilizar `ifconfig`:

```
$ ifconfig enp0s9:sub1 172.16.15.16/16
```

Respuestas a los ejercicios de exploración

1. ¿Qué subcomando de `ip` se puede utilizar para configurar el etiquetado vlan?

   `ip link` tiene una opción de `vlan` que puede ser utilizada. A continuación se muestra un ejemplo de etiquetado de una subinterfaz con vlan 20.

   ```
   # ip link add link enp0s9 name enp0s9.20 type vlan id 20
   ```

2. ¿Cómo se configura una ruta por defecto?

   Utilizando `route` o `ip route`:

   ```
   # route add default gw 192.168.1.1
   # ip route add default via 192.168.1.1
   ```

3. ¿Cómo se puede obtener información detallada sobre el comando `ip neighbour`? ¿Qué sucede si lo ejecuta por sí mismo?

   Leyendo la página de man:

   ```
   $ man ip-neigbour
   ```

   Se muestra la caché ARP:

   ```
   $ ip neighbour
   10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
   ```

4. ¿Cómo se hace una copia de seguridad de la tabla de enrutamiento? ¿Cómo se restaura desde ella?

   El siguiente ejemplo muestra la copia de seguridad y la restauración de una tabla de enrutamiento:

   ```
   # ip route save > /root/routes/route_backup
   # ip route restore < /root/routes/route_backup
   ```

5. ¿Qué subcomando `ip` puede utilizarse para configurar las opciones del árbol de expansión?

   De forma similar a la gestión de la configuración de las vlan, `ip link` puede configurar el árbol de expansión utilizando el tipo `bridge`. El ejemplo muestra la adición de una interfaz virtual con una prioridad STP de 50:

   ```
   # ip link add link enp0s9 name enp0s9.50 type bridge priority 50
   ```




109.3 Lección 2

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.3 Resolución de problemas básicos de red
Lección:	2 de 2

Introducción

Los sistemas operativos basados en Linux tienen una variedad de herramientas para solucionar problemas de red. Esta lección va a cubrir algunas de las más comunes. En este punto debería tener un conocimiento de los modelos OSI u otros modelos de red, el direccionamiento IPv4 o IPv6, y los fundamentos del enrutamiento y la conmutación.

La mejor manera de probar una conexión de red es intentar utilizar su aplicación. Cuando eso no funciona, hay muchas herramientas disponibles para ayudar a diagnosticar el problema.

Probar las conexiones con `ping`

Los comandos `ping` y `ping6` pueden utilizarse para enviar una solicitud de eco ICMP a una dirección IPv4 o IPv6, respectivamente. Una petición de eco ICMP envía una pequeña cantidad de datos a la dirección de destino. Si la dirección de destino es alcanzable, enviará un mensaje de respuesta de eco ICMP de vuelta al remitente con los mismos datos que le fueron enviados:

```
$ ping -c 3 192.168.50.2
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=64 time=0.525 ms
64 bytes from 192.168.50.2: icmp_seq=2 ttl=64 time=0.419 ms
64 bytes from 192.168.50.2: icmp_seq=3 ttl=64 time=0.449 ms

--- 192.168.50.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.419/0.464/0.525/0.047 ms
```

```
$ ping6 -c 3 2001:db8::10
PING 2001:db8::10(2001:db8::10) 56 data bytes
64 bytes from 2001:db8::10: icmp_seq=1 ttl=64 time=0.425 ms
64 bytes from 2001:db8::10: icmp_seq=2 ttl=64 time=0.480 ms
64 bytes from 2001:db8::10: icmp_seq=3 ttl=64 time=0.725 ms

--- 2001:db8::10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.425/0.543/0.725/0.131 ms
```

La opción `-c` se utiliza para especificar el número de paquetes a enviar. Si omite esta opción, `ping` y `ping6` continuarán enviando paquetes hasta que lo detengas, normalmente con la combinación de teclado `Ctrl` + `C`.

Que no pueda hacer ping a un host no significa que no pueda conectarse a él. Muchas organizaciones tienen cortafuegos o listas de control de acceso al router que bloquean todo lo que no sea el mínimo necesario para que sus sistemas funcionen. Esto incluye las peticiones y respuestas de eco ICMP. Dado que estos paquetes pueden incluir datos arbitrarios, un atacante inteligente podría utilizarlos para exfiltrar datos.

Traceroute

Los programas `traceroute` y `traceroute6` pueden utilizarse para mostrar la ruta que sigue un paquete para llegar a su destino. Lo hacen enviando múltiples paquetes al destino, incrementando el campo *Time-To-Live* (TTL) de la cabecera IP con cada paquete subsiguiente. Cada router a lo largo del camino responderá con un mensaje ICMP de TTL excedido:

```
$ traceroute 192.168.1.20
traceroute to 192.168.1.20 (192.168.1.20), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.396 ms  0.171 ms  0.132 ms
 2  192.168.1.20 (192.168.1.20)  2.665 ms  2.573 ms  2.573 ms
$ traceroute 192.168.50.2
traceroute to 192.168.50.2 (192.168.50.2), 30 hops max, 60 byte packets
 1  192.168.50.2 (192.168.50.2)  0.433 ms  0.273 ms  0.171 ms
$ traceroute6 2001:db8::11
traceroute to 2001:db8::11 (2001:db8::11), 30 hops max, 80 byte packets
 1  2001:db8::11 (2001:db8::11)  0.716 ms  0.550 ms  0.641 ms
$ traceroute 2001:db8::11
traceroute to 2001:db8::11 (2001:db8::11), 30 hops max, 80 byte packets
 1  2001:db8::10 (2001:db8::11)  0.617 ms  0.461 ms  0.387 ms
$ traceroute net2.example.net
traceroute to net2.example.net (192.168.50.2), 30 hops max, 60 byte packets
 1  net2.example.net (192.168.50.2)  0.533 ms  0.529 ms  0.504 ms
$ traceroute6 net2.example.net
traceroute to net2.example.net (2001:db8::11), 30 hops max, 80 byte packets
 1  net2.example.net (2001:db8::11)  0.738 ms  0.607 ms  0.304 ms
```

Por defecto, `traceroute` envía 3 paquetes UDP con datos basura al puerto 33434, incrementándolo cada vez que envía un paquete. Cada línea de la salida del comando es una interfaz de router por la que atraviesa el paquete. El tiempo mostrado en cada línea de la salida es el tiempo de ida y vuelta de cada paquete. La dirección IP es la dirección de la interfaz del router en cuestión. Si `traceroute` puede, utiliza el nombre DNS de la interfaz del router. A veces verá `*` en lugar de un tiempo. Cuando esto sucede, significa que `traceroute` nunca recibió el mensaje de TTL excedido para este paquete. Cuando se empieza a ver esto, suele indicar que la última respuesta es el último salto de la ruta.

Si tiene acceso a `root`, la opción `-I` hará que `traceroute` utilice peticiones de eco ICMP en lugar de paquetes UDP. Esto suele ser más efectivo que el UDP porque es más probable que el host de destino responda a una petición de eco ICMP que al paquete UDP:

```
# traceroute -I learning.lpi.org
traceroute to learning.lpi.org (208.94.166.201), 30 hops max, 60 byte packets
 1  047-132-144-001.res.spectrum.com (47.132.144.1)  9.764 ms  9.702 ms  9.693 ms
 2  096-034-094-106.biz.spectrum.com (96.34.94.106)  8.389 ms  8.481 ms  8.480 ms
 3  dtr01hlrgnc-gbe-4-15.hlrg.nc.charter.com (96.34.64.172)  8.763 ms  8.775 ms
8.770 ms
 4  acr01mgtnnc-vln-492.mgtn.nc.charter.com (96.34.67.202)  27.080 ms  27.154 ms
27.151 ms
 5  bbr01gnvlsc-bue-3.gnvl.sc.charter.com (96.34.2.112)  31.339 ms  31.398 ms
31.395 ms
 6  bbr01aldlmi-tge-0-0-0-13.aldl.mi.charter.com (96.34.0.161)  39.092 ms  38.794
ms  38.821 ms
 7  prr01ashbva-bue-3.ashb.va.charter.com (96.34.3.51)  34.208 ms  36.474 ms
36.544 ms
 8  bx2-ashburn.bell.ca (206.126.236.203)  53.973 ms  35.975 ms  38.250 ms
 9  tcore4-ashburnbk_0-12-0-0.net.bell.ca (64.230.125.190)  66.315 ms  65.319 ms
65.345 ms
10  tcore4-toronto47_2-8-0-3.net.bell.ca (64.230.51.22)  67.427 ms  67.502 ms
67.498 ms
11  agg1-toronto47_xe-7-0-0_core.net.bell.ca (64.230.161.114)  61.270 ms  61.299 ms
61.291 ms
12  dis4-clarkson16_5-0.net.bell.ca (64.230.131.98)  61.101 ms  61.177 ms  61.168
ms
13  207.35.12.142 (207.35.12.142)  70.009 ms  70.069 ms  59.893 ms
14  unassigned-117.001.centrilogic.com (66.135.117.1)  61.778 ms  61.950 ms  63.041
ms
15  unassigned-116.122.akn.ca (66.135.116.122)  62.702 ms  62.759 ms  62.755 ms
16  208.94.166.201 (208.94.166.201)  62.936 ms  62.932 ms  62.921 ms
```

Algunas organizaciones bloquean las peticiones y respuestas de eco ICMP. Para evitarlo, puede utilizar TCP. Al utilizar un puerto TCP abierto conocido, puede garantizar que el host de destino responderá. Para usar TCP, utilice la opción `-T` junto con `-p` para especificar el puerto. Al igual que con las peticiones de eco ICMP, debe tener acceso a `root` para hacer esto:


```
# traceroute -m 60 -T -p 80  learning.lpi.org
traceroute to learning.lpi.org (208.94.166.201), 60 hops max, 60 byte packets
 1  * * *
 2  096-034-094-106.biz.spectrum.com (96.34.94.106)  12.178 ms  12.229 ms  12.175
ms
 3  dtr01hlrgnc-gbe-4-15.hlrg.nc.charter.com (96.34.64.172)  12.134 ms  12.093 ms
12.062 ms
 4  acr01mgtnnc-vln-492.mgtn.nc.charter.com (96.34.67.202)  31.146 ms  31.192 ms
31.828 ms
 5  bbr01gnvlsc-bue-3.gnvl.sc.charter.com (96.34.2.112)  39.057 ms  46.706 ms
39.745 ms
 6  bbr01aldlmi-tge-0-0-0-13.aldl.mi.charter.com (96.34.0.161)  50.590 ms  58.852
ms  58.841 ms
 7  prr01ashbva-bue-3.ashb.va.charter.com (96.34.3.51)  34.556 ms  37.892 ms
38.274 ms
 8  bx2-ashburn.bell.ca (206.126.236.203)  38.249 ms  36.991 ms  36.270 ms
 9  tcore4-ashburnbk_0-12-0-0.net.bell.ca (64.230.125.190)  66.779 ms  63.218 ms
tcore3-ashburnbk_100ge0-12-0-0.net.bell.ca (64.230.125.188)  60.441 ms
10  tcore4-toronto47_2-8-0-3.net.bell.ca (64.230.51.22)  63.932 ms  63.733 ms
68.847 ms
11  agg2-toronto47_xe-7-0-0_core.net.bell.ca (64.230.161.118)  60.144 ms  60.443 ms
agg1-toronto47_xe-7-0-0_core.net.bell.ca (64.230.161.114)  60.851 ms
12  dis4-clarkson16_5-0.net.bell.ca (64.230.131.98)  67.246 ms dis4-clarkson16_7-
0.net.bell.ca (64.230.131.102)  68.404 ms dis4-clarkson16_5-0.net.bell.ca
(64.230.131.98)  67.403 ms
13  207.35.12.142 (207.35.12.142)  66.138 ms  60.608 ms  64.656 ms
14  unassigned-117.001.centrilogic.com (66.135.117.1)  70.690 ms  62.190 ms  61.787
ms
15  unassigned-116.122.akn.ca (66.135.116.122)  62.692 ms  69.470 ms  68.815 ms
16  208.94.166.201 (208.94.166.201)  61.433 ms  65.421 ms  65.247 ms
17  208.94.166.201 (208.94.166.201)  64.023 ms  62.181 ms  61.899 ms
```

Al igual que `ping`, `traceroute` tiene sus limitaciones. Es posible que los cortafuegos y los routers bloqueen los paquetes enviados o devueltos por `traceroute`. Si tiene acceso `root`, hay opciones que pueden ayudare a obtener resultados precisos.

Búsqueda de MTU con `tracepath`

El comando `tracepath` es similar a `traceroute`. La diferencia es que rastrea los tamaños de las *Unidades Máximas de Transmisión* (MTU) a lo largo de la ruta. MTU es un ajuste configurado en una interfaz de red o una limitación de hardware de la unidad de datos de protocolo más grande que puede transmitir o recibir. El programa `tracepath` funciona de la misma manera que `traceroute` en el sentido de que incrementa el TTL con cada paquete. Se diferencia en que envía un datagrama UDP

muy grande. Es casi inevitable que el datagrama sea más grande que el dispositivo con la MTU más pequeña de la ruta. Cuando el paquete llega a este dispositivo, éste suele responder con un paquete de destino inalcanzable. El paquete ICMP de destino inalcanzable tiene un campo para la MTU del enlace por el que enviaría el paquete si pudiera. Tracepath envía entonces todos los paquetes subsiguientes con este tamaño:

```
$ tracepath 192.168.1.20
 1?: [LOCALHOST]                                          pmtu 1500
 1:  10.0.2.2                                               0.321ms
 1:  10.0.2.2                                               0.110ms
 2:  192.168.1.20                                           2.714ms reached
     Resume: pmtu 1500 hops 2 back 64
```

A diferencia de `traceroute`, debe utilizar explícitamente `tracepath6` para IPv6:

```
$ tracepath 2001:db8::11
tracepath: 2001:db8::11: Address family for hostname not supported
$ tracepath6 2001:db8::11
 1?: [LOCALHOST]                        0.027ms pmtu 1500
 1:  net2.example.net                                       0.917ms reached
 1:  net2.example.net                                       0.527ms reached
     Resume: pmtu 1500 hops 1 back 1
```

La salida es similar a `traceroute`. La ventaja de `tracepath` es que en la última línea muestra la MTU más pequeña de todo el enlace. Esto puede ser útil para solucionar problemas de conexiones que no pueden manejar fragmentos.

Al igual que con las herramientas de solución de problemas anteriores, existe la posibilidad de que los equipos bloqueen sus paquetes.

Crear conexiones arbitrarias

El programa `nc`, conocido como netcat, puede enviar o recibir datos arbitrarios a través de una conexión de red TCP o UDP. Los siguientes ejemplos deberían dejar clara su funcionalidad.

A continuación se muestra un ejemplo de configuración de un listener en el puerto `1234`:

```
$ nc -l 1234
LPI Example
```

La salida de `LPI Example` aparece después del siguiente ejemplo, que está configurando un remitente netcat para enviar paquetes a `net2.example.net` en el puerto `1234`. La opción `-l` se utiliza para especificar que se desea que `nc` reciba datos en lugar de enviarlos:

```
$ nc net2.example.net 1234
LPI Example
```

Pulse `Ctrl` + `C` en cualquiera de los dos sistemas para detener la conexión.

Netcat funciona tanto con direcciones IPv4 como IPv6. Funciona tanto con TCP como con UDP. Incluso se puede utilizar para configurar un burdo shell remoto.

WARNING	Tenga en cuenta que no todas las instalaciones de `nc` soportan la opción `-e`. Asegúrese de revisar las páginas de manual de su instalación para obtener información de seguridad sobre esta opción, así como métodos alternativos para ejecutar comandos en un sistema remoto.

```
$ hostname
net2
$ nc -u -e /bin/bash -l 1234
```

La opción `-u` es para UDP. La opción `-e` indica a netcat que envíe todo lo que recibe a la entrada estándar del ejecutable que le sigue. En este ejemplo: `/bin/bash`.

```
$ hostname
net1
$ nc -u net2.example.net 1234
hostname
net2
pwd
/home/emma
```

¿Nota que la salida del comando `hostname` coincide con la del host que escucha y la salida del comando `pwd` con la de un directorio?

Ver conexiones y oyentes actuales

Los programas `netstat` y `ss` pueden utilizarse para ver el estado de sus oyentes y conexiones actuales. Al igual que `ifconfig`, `netstat` es una herramienta heredada. Tanto `netstat` como `ss`

tienen salidas y opciones similares. Aquí están algunas opciones disponibles para ambos programas:

`-a`
: Muestra todos los sockets.

`-l`
: Presenta los sockets en escucha.

`-p`
: Muestra el proceso asociado a la conexión.

`-n`
: Evita la búsqueda de nombres tanto para los puertos como para las direcciones.

`-t`
: Presenta las conexiones TCP.

`-u`
: Muestra las conexiones UDP.

Los ejemplos siguientes muestran la salida de un conjunto de opciones comúnmente utilizadas para ambos programas:


```
# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
892/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
1141/master
tcp6       0      0 :::22                   :::*                    LISTEN
892/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN
1141/master
udp        0      0 0.0.0.0:68              0.0.0.0:*
692/dhclient
# ss -tulnp
# ss -tulnp
Netid  State      Recv-Q Send-Q      Local Address:Port                     Peer
Address:Port
udp    UNCONN     0      0                          :68
*:                     users:(("dhclient",pid=693,fd=6))
tcp    LISTEN     0      128                        :22
*:                     users:(("sshd",pid=892,fd=3))
tcp    LISTEN     0      100                127.0.0.1:25
:                     users:(("master",pid=1099,fd=13))
tcp    LISTEN     0      128                     [::]:22
[::]:*                     users:(("sshd",pid=892,fd=4))
tcp    LISTEN     0      100                    [::1]:25
[::]:*                     users:(("master",pid=1099,fd=14))
```

La columna `Recv-Q` es el número de paquetes que un socket ha recibido pero no ha pasado a su programa. La columna `Send-Q` es el número de paquetes que un socket ha enviado y que no han sido reconocidos por el receptor. El resto de las columnas se explican por sí mismas.

Ejercicios guiados

1. ¿Qué comando(s) utilizaría para enviar un eco ICMP a `learning.lpi.org`?

2. ¿Cómo podría determinar la ruta a `8.8.8.8`?

3. ¿Qué comando le mostraría si algún proceso está escuchando en el puerto TCP 80?

4. ¿Cómo se puede saber qué proceso está escuchando en un puerto?

5. ¿Cómo se puede determinar la MTU máxima de una ruta de red?

Ejercicios de exploración

1. ¿Cómo podría utilizar netcat para enviar una petición HTTP a un servidor web?

2. ¿Cuáles son algunas de las razones por las que puede fallar el ping a un host?

3. Nombre una herramienta que pueda utilizar para ver los paquetes de red que llegan o salen de un host Linux.

4. ¿Cómo se puede forzar a `traceroute` a utilizar una interfaz diferente?

5. ¿Es posible que `traceroute` informe de las MTU?

Resumen

La red se configura normalmente mediante los scripts de inicio del sistema o un ayudante como NetworkManager. La mayoría de las distribuciones tienen herramientas que editarán los archivos de configuración de los scripts de inicio por usted. Consulte la documentación de su distribución para más detalles.

Ser capaz de configurar manualmente la red le permite solucionar los problemas de forma más eficaz. Es útil en entornos mínimos utilizados para cosas como la restauración de copias de seguridad o la migración a un nuevo hardware.

Las utilidades que se cubren en esta sección tienen más funcionalidades que las ya anteriormente mencionadas en la lección. Valdría la pena hojear la página de manual de cada una para familiarizarse con las opciones disponibles. Los comandos `ss` e `ip` son herramientas modernas y recomendadas, mientras que el resto de las que se cubren, aunque todavía son de uso común, se consideran herramientas heredadas.

La mejor manera de familiarizarse con las herramientas cubiertas es la práctica. Utilizando un ordenador con una cantidad modesta de RAM, es posible configurar un laboratorio de red virtual utilizando máquinas virtuales con las que se puede practicar. Tres máquinas virtuales son suficientes para familiarizarse con las herramientas enumeradas.

Los comandos cubiertos en esta lección incluyen:

`ping` o **`ping6`**
: Se utilizan para transmitir paquetes ICMP a un host remoto para probar la disponibilidad de una conexión de red.

`traceroute` y **`traceroute6`**
: Se emplean para trazar una ruta a través de una red para determinar la conectividad de la misma.

`tracepath` y **`tracepath6`**
: Se usan para trazar una ruta a través de una red, así como para determinar los tamaños de MTU a lo largo de una ruta.

`nc`
: Se utiliza para establecer conexiones arbitrarias en una red para probar la conectividad, así como para consultar una red en busca de servicios y dispositivos disponibles.

`netstat`
: Comando heredado utilizado para determinar las conexiones de red abiertas y las estadísticas de un sistema.

`ss`
: Comando moderno utilizado para determinar las conexiones de red abiertas y las estadísticas de un sistema.

Respuestas a los ejercicios guiados

1. ¿Qué comando(s) utilizarías para enviar un eco ICMP a `learning.lpi.org`?

   Utilizaría `ping` o `ping6`:

   ```
   $ ping learning.lpi.org
   ```

   o

   ```
   $ ping6 learning.lpi.org
   ```

2. ¿Cómo podría determinar la ruta a `8.8.8.8`?

   Utilizando los comandos `tracepath` o `traceroute`.

   ```
   $ tracepath 8.8.8.8
   ```

   o

   ```
   $ traceroute 8.8.8.8
   ```

3. ¿Qué comando le mostraría si algún proceso está escuchando en el puerto TCP 80?

   Con `ss`:

   ```
   $ ss -ln | grep ":80"
   ```

   Con `netstat`:

   ```
   $ netstat -ln | grep ":80"
   ```

   Aunque no figura como requisito para el examen, también puede utilizar `lsof`:

   ```
   # lsof -Pi:80
   ```


4. ¿Cómo se puede saber qué proceso está escuchando en un puerto?

   De nuevo, hay múltiples maneras de hacer esto. Puede usar `lsof` de la misma manera que la respuesta anterior, reemplazando el número de puerto. También puede usar `netstat` o `ss` con la opción `-p`. Recuerde que `netstat` se considera una herramienta heredada.

   ```
   # netstat -lnp | grep ":22"
   ```

   Las mismas opciones que funcionan con `netstat` también funcionan con `ss`:

   ```
   # ss -lnp | grep ":22"
   ```

5. ¿Cómo se puede determinar la MTU máxima de una ruta de red?

   Utilizando el comando `tracepath`:

   ```
   $ tracepath somehost.example.com
   ```

Respuestas a los ejercicios de exploración

1. ¿Cómo podrías utilizar netcat para enviar una petición HTTP a un servidor web?

   Introduciendo en el terminal la línea de petición HTTP, cualquier cabecera y una línea en blanco:

```
$ nc learning.lpi.org 80
GET /index.html HTTP/1.1
HOST: learning.lpi.org

HTTP/1.1 302 Found
Location: https://learning.lpi.org:443/index.html
Date: Wed, 27 May 2020 22:54:46 GMT
Content-Length: 5
Content-Type: text/plain; charset=utf-8

Found
```

2. ¿Cuáles son algunas de las razones por las que puede fallar el ping a un host?

   Hay varias razones posibles. Éstas son algunas:

   - El host remoto no funciona.
   - Un ACL del router está bloqueando el ping.
   - El firewall del host remoto está bloqueando el ping.
   - Puede que esté utilizando un nombre o una dirección de host incorrecta.
   - Su resolución de nombres está devolviendo una dirección incorrecta.
   - La configuración de red de su máquina es incorrecta.
   - El cortafuegos de su máquina lo está bloqueando.
   - La configuración de red del host remoto es incorrecta.
   - La(s) interfaz(es) de su máquina está(n) desconectada(s).
   - La(s) interfaz(es) de la máquina remota está(n) desconectada(s).
   - Un componente de la red, como un conmutador, un cable o un router entre su máquina y la remota, ya no funciona.

3. Nombre una herramienta que pueda utilizar para ver los paquetes de red que llegan o salen de un host Linux.

Se pueden utilizar tanto `tcpdump` como `wireshark`.

4. ¿Cómo se puede forzar a `traceroute` a utilizar una interfaz diferente?

   Utilizando la opción `-i`:

```
$ traceroute -i eth2 learning.lpi.org
traceroute -i eth2 learning.lpi.org
traceroute to learning.lpi.org (208.94.166.201), 30 hops max, 60 byte packets
...
```

5. ¿Es posible que `traceroute` informe de las MTU?

   Sí, con la opción `--mtu`:

```
# traceroute -I --mtu  learning.lpi.org
traceroute to learning.lpi.org (208.94.166.201), 30 hops max, 65000 byte
packets
 1  047-132-144-001.res.spectrum.com (47.132.144.1)  9.974 ms F=1500  10.476 ms
4.743 ms
 2  096-034-094-106.biz.spectrum.com (96.34.94.106)  8.697 ms  9.963 ms  10.321
ms
...
```



109.4 Configuración DNS en el lado del cliente

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 109.4

Importancia

2

Áreas de conocimiento clave

- Consultar servidores DNS remotos.
- Configurar la resolución de nombres local y usar servidores DNS remotos.
- Modificar el orden en que se realiza la resolución de nombres.
- Depurar errores relacionados con la resolución de nombres.
- Conocimientos de `systemd-resolved`

Lista parcial de archivos, términos y utilidades

- `/etc/hosts`
- `/etc/resolv.conf`
- `/etc/nsswitch.conf`
- `host`
- `dig`
- `getent`



109.4 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	109 Fundamentos de redes
Objetivo:	109.4 Configuración DNS en el lado del cliente
Lección:	1 de 1

Introducción

Esta lección cubre la configuración de la resolución de nombres del lado del cliente y cómo utilizar algunas herramientas de resolución de nombres de la CLI.

Recordar y mantener direcciones IP, UID, GID y otros números para todo no es factible. Los servicios de resolución de nombres traducen nombres fáciles de recordar a números y viceversa. Esta lección se centra en la resolución de nombres de host, pero un proceso similar ocurre para los nombres de usuario, los nombres de grupo, los números de puerto y varios otros.

Proceso de resolución de nombres

Los programas que resuelven nombres a números casi siempre utilizan funciones proporcionadas por la biblioteca estándar de C, que en los sistemas Linux es el glibc del proyecto GNU. Lo primero que hacen estas funciones es leer el archivo `/etc/nsswitch.conf` para obtener instrucciones sobre cómo resolver ese tipo de nombre. Esta lección se centra en la resolución de nombres de host, pero el mismo proceso se aplica también a otros tipos de resolución de nombres. Una vez que el proceso lee `/etc/nsswitch.conf`, busca el nombre de la manera especificada. Dado que `/etc/nsswitch.conf` soporta plugins, lo que viene a continuación puede ser cualquier cosa. Una vez que la función ha

terminado de buscar el nombre o el número, devuelve el resultado al proceso que lo ha llamado.

Clases de DNS

El DNS tiene tres clases de registros, IN, HS y CH. En esta lección, todas las consultas DNS serán del tipo IN. La clase IN es para las direcciones de Internet que utilizan la pila TCP/IP. CH es para ChaosNet, que es una tecnología de red que tuvo una corta vida y ya no está en uso. La clase HS es para Hesiod. Hesiod es una forma de almacenar cosas como passwd y entradas de grupo en DNS. Hesiod está más allá del alcance de esta lección.

Comprendiendo el archivo `/etc/nsswitch.conf`

La mejor manera de aprender sobre este archivo es leer la página man que forma parte del proyecto Linux man-pages. Está disponible en la mayoría de los sistemas. Se puede acceder a ella con el comando `man nsswitch.conf`. También se puede encontrar en https://man7.org/linux/man-pages/dir_section_5.html

A continuación se muestra un ejemplo sencillo de `/etc/nsswitch.conf` de su página man:

```
            passwd:         compat
            group:          compat
            shadow:         compat

            hosts:          dns [!UNAVAIL=return] files
            networks:       nis [NOTFOUND=return] files
            ethers:         nis [NOTFOUND=return] files
            protocols:      nis [NOTFOUND=return] files
            rpc:            nis [NOTFOUND=return] files
            services:       nis [NOTFOUND=return] files
            # This is a comment. It is ignored by the resolution functions.
```

El archivo está organizado en columnas. La columna del extremo izquierdo es el tipo de base de datos de nombres. El resto de las columnas son los métodos que las funciones de resolución deben utilizar para buscar un nombre. Los métodos van seguidos de las funciones de izquierda a derecha. Las columnas con `[]` se utilizan para proporcionar alguna lógica condicional limitada a la columna inmediatamente a la izquierda de la misma.

Supongamos que un proceso intenta resolver el nombre de host `learning.lpi.org`. Haría una llamada a la biblioteca C apropiada (probablemente `gethostbyname`). Esta función leerá entonces `/etc/nsswitch.conf`. Como el proceso está buscando un nombre de host, encontrará la línea que comienza con `hosts`. Entonces intentará usar el DNS para resolver el nombre. La siguiente columna,

`[!UNAVAIL=return]` significa que si el servicio *no* está disponible, entonces no intente la siguiente fuente, es decir, si DNS está disponible, deje de intentar resolver el nombre del host aunque los servidores de nombres no puedan. Si el DNS no está disponible, entonces continúe con la siguiente fuente. En este caso, la siguiente fuente es `files`.

Cuando veas una columna con el formato `[resultado=action]`, significa que la búsqueda del resolver de la columna a la izquierda es `resultado`, entonces se realiza `acción`. Si `resultado` es similar a un `!`, significa que el resultado no es igual a este, entonces se realiza `acción`. Para las descripciones de los posibles resultados y acciones, consulte la página de man.

Ahora supongamos que un proceso está tratando de resolver un número de puerto a un nombre de servicio. Para ello, leerá la línea `servicios`. La primera fuente listada es NIS. Estas siglas significa *Network Information Service* (a veces se le llama yellow pages). Es un antiguo servicio que permitía la gestión centralizada de objetos como los usuarios. Ya no se usa mucho debido a su escasa seguridad. La siguiente columna `[NOTFOUND=return]` significa que si la búsqueda tiene éxito pero no encontra el servicio, debe dejar de buscar. Si la condición mencionada no se aplica, utilice los archivos locales.

Cualquier cosa a la derecha de `#` es un comentario y se ignora.

El archivo `/etc/resolv.conf`.

El fichero `/etc/resolv.conf` se utiliza para configurar la resolución de hosts mediante DNS. Algunas distribuciones tienen scripts de inicio, demonios y otras herramientas que escriben en este archivo. Tenga esto en cuenta cuando edite manualmente este archivo. Compruebe su distribución y la documentación de cualquier herramienta de configuración de red si este es el caso. Algunas herramientas, como Network Manager, dejarán un comentario en el archivo para informarle que los cambios manuales se sobrescribirán.

Al igual que con `/etc/nsswitch.conf`, hay una página man asociada al fichero. Se puede acceder a ella con el comando `man resolv.conf` o en https://man7.org/linux/man-pages/man5/resolv.conf.5.html.

El formato del archivo es bastante sencillo. En la columna de la izquierda se encuentra la opción `name`. El resto de las columnas en la misma línea son el valor de la opción.

La opción más común es la del `nameserver`. Se utiliza para especificar la dirección IPv4 o IPv6 de un servidor DNS. A partir de la fecha de este escrito, puede especificar hasta tres servidores de nombres. Si su `/etc/resolv.conf` no tiene la opción `nameserver`, su sistema utilizará por defecto el servidor de nombres de la máquina local.

A continuación se muestra un archivo de ejemplo simple que es representativo de las configuraciones más comunes:

```
search lpi.org
nameserver 10.0.0.53
nameserver fd00:ffff::2:53
```

La opción `search` se utiliza para permitir las búsquedas de forma corta. En el ejemplo, se ha configurado un único dominio de búsqueda de `lpi.org`. Esto significa que cualquier intento de resolver un nombre de host sin una porción de dominio tendrá `.lpi.org` añadido antes de la búsqueda. Por ejemplo, si intentara buscar un host llamado `aprendizaje`, el resolvedor buscaría `aprendizaje.lpi.org`. Puede tener configurados hasta seis dominios de búsqueda.

Otra opción común es la opción `domain`. Se utiliza para establecer el nombre de dominio local. Si esta opción no está presente, se utiliza por defecto todo lo que sigue al primer `.` en el nombre de host de la máquina. Si el nombre de host no contiene un `.`, se asume que la máquina es parte del dominio raíz. Al igual que `search`, `domain` puede utilizarse para búsquedas de nombres cortos.

Tenga en cuenta que `domain` y `search` son mutuamente excluyentes. Si ambos están presentes, se utiliza la última instancia del archivo.

Hay varias opciones que se pueden establecer para afectar el comportamiento del resolver. Para establecerlas, utilice la palabra clave `options`, seguida del nombre de la opción a establecer, y si es aplicable, un `:` seguido del valor. A continuación se muestra un ejemplo de configuración de la opción de tiempo de espera, que es el tiempo en segundos que el resolvedor esperará a un servidor de nombres antes de rendirse:

```
option timeout:3
```

Hay otras opciones en `resolv.conf`, pero estas son las más comunes.

El archivo `/etc/hosts`

El archivo `/etc/hosts` se utiliza para resolver nombres a direcciones IP y viceversa. Tanto IPv4 como IPv6 son compatibles. La columna de la izquierda es la dirección IP, el resto son nombres asociados a esa dirección. El uso más común de `/etc/hosts` es para hosts y direcciones donde el DNS no es posible, como las direcciones de loopback. En el ejemplo siguiente, se definen las direcciones IP de los componentes críticos de la infraestructura.

Este es un ejemplo realista de un archivo `/etc/hosts`:

```
127.0.0.1       localhost
127.0.1.1       proxy
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.0.0.1        gateway.lpi.org gateway gw
fd00:ffff::1    gateway.lpi.org gateway gw

10.0.1.53       dns1.lpi.org
fd00:ffff::1:53 dns1.lpi.org
10.0.2.53       dns2.lpi.org
fd00:ffff::2:53 dns2.lpi.org
```

systemd-resolved

Systemd proporciona un servicio llamado `systemd-resolved`. Proporciona mDNS, DNS y LLMNR. Cuando se ejecuta, escucha las peticiones DNS en `127.0.0.53`. No proporciona un servidor DNS completo. Las peticiones DNS que recibe se buscan consultando los servidores configurados en `/etc/systemd/resolv.conf` o `/etc/resolv.conf`. Si desea utilizar esto, use `resolve` para `hosts` en `/etc/nsswitch.conf`. Tenga en cuenta que el paquete del sistema operativo que tiene la biblioteca `systemd-resolved` puede no estar instalado por defecto.

Herramientas de resolución de nombres

Hay muchas herramientas disponibles para los usuarios de Linux para la resolución de nombres. Esta lección cubre tres. Una, `getent`, es útil para ver cómo se resuelven las peticiones del mundo real. Otro comando es `host`, que es útil para consultas DNS simples. Un programa llamado `dig` es útil para operaciones DNS complejas que pueden ayudar a solucionar problemas del servidor DNS.

El comando `getent`

La utilidad `getent` se utiliza para mostrar las entradas de las bases de datos del servicio de nombres. Puede recuperar registros de cualquier fuente configurable por `/etc/nsswitch.conf`.

Para utilizar `getent`, siga el comando con el tipo de nombre que desea resolver y, opcionalmente, una entrada específica para buscar. Si sólo especifica el tipo de nombre, `getent` intentará mostrar todas las entradas de ese tipo de datos:

```
$ getent hosts
127.0.0.1       localhost
127.0.1.1       proxy
10.0.1.53       dns1.lpi.org
10.0.2.53       dns2.lpi.org
127.0.0.1       localhost ip6-localhost ip6-loopback
$ getent hosts dns1.lpi,org
fd00:ffff::1:53 dns1.lpi.org
```

A partir de la versión 2.2.5 de glibc, puede forzar a `getent` a utilizar una fuente de datos específica con la opción `-s`. El siguiente ejemplo lo demuestra:

```
$ getent -s files hosts learning.lpi.org
::1             learning.lpi.org
$ getent -s dns hosts learning.lpi.org
208.94.166.198  learning.lpi.org
```

El comando `host`

`host` es un programa sencillo para buscar entradas DNS. Sin opciones, si a `host` se le da un nombre, devuelve los conjuntos de registros A, AAAA y MX. Si se le da una dirección IPv4 o IPv6, devuelve el registro PTR si hay uno disponible:

```
$ host wikipedia.org
wikipedia.org has address 208.80.154.224
wikipedia.org has IPv6 address 2620:0:861:ed1a::1
wikipedia.org mail is handled by 10 mx1001.wikimedia.org.
wikipedia.org mail is handled by 50 mx2001.wikimedia.org.
$ host 208.80.154.224
224.154.80.208.in-addr.arpa domain name pointer text-lb.eqiad.wikimedia.org.
```

Si busca un tipo de registro específico, puede utilizar `host -t`:

```
$ host -t NS lpi.org
lpi.org name server dns1.easydns.com.
lpi.org name server dns3.easydns.ca.
lpi.org name server dns2.easydns.net.
$ host -t SOA lpi.org
lpi.org has SOA record dns1.easydns.com. zone.easydns.com. 1593109612 3600 600
1209600 300
```

`host` también puede utilizarse para consultar un servidor de nombres específico si no desea utilizar los que se encuentran en `/etc/resolv.conf`. Simplemente añada la dirección IP o el nombre del servidor que desea utilizar como último argumento:

```
$ host -t MX lpi.org dns1.easydns.com
Using domain server:
Name: dns1.easydns.com
Address: 64.68.192.10#53
Aliases:

lpi.org mail is handled by 10 aspmx4.googlemail.com.
lpi.org mail is handled by 10 aspmx2.googlemail.com.
lpi.org mail is handled by 5 alt1.aspmx.l.google.com.
lpi.org mail is handled by 0 aspmx.l.google.com.
lpi.org mail is handled by 10 aspmx5.googlemail.com.
lpi.org mail is handled by 10 aspmx3.googlemail.com.
lpi.org mail is handled by 5 alt2.aspmx.l.google.com.
```

El comando `dig`

Otra herramienta para consultar servidores DNS es `dig`. Este comando es mucho más detallado que `host`. Por defecto, `dig` consulta los registros A. Probablemente es demasiado tedioso buscar una dirección IP o un nombre de host. `dig` funcionará para búsquedas sencillas, pero es más adecuado para solucionar problemas de configuración del servidor DNS:

```
$ dig learning.lpi.org

; <<>> DiG 9.11.5-P4-5.1+deb10u1-Debian <<>> learning.lpi.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63004
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ca7a415be1cec45592b082665ef87f3483b81ddd61063c30 (good)
;; QUESTION SECTION:
;learning.lpi.org.      IN  A

;; ANSWER SECTION:
learning.lpi.org.   600 IN  A   208.94.166.198

;; AUTHORITY SECTION:
lpi.org.        86400   IN  NS  dns2.easydns.net.
lpi.org.        86400   IN  NS  dns1.easydns.com.
lpi.org.        86400   IN  NS  dns3.easydns.ca.

;; ADDITIONAL SECTION:
dns1.easydns.com.   172682  IN  A   64.68.192.10
dns2.easydns.net.   170226  IN  A   198.41.222.254
dns1.easydns.com.   172682  IN  AAAA    2400:cb00:2049:1::a29f:1835
dns2.easydns.net.   170226  IN  AAAA    2400:cb00:2049:1::c629:defe

;; Query time: 135 msec
;; SERVER: 192.168.1.20#53(192.168.1.20)
;; WHEN: Sun Jun 28 07:29:56 EDT 2020
;; MSG SIZE  rcvd: 266
```

Como puede ver, `dig` proporciona mucha información. La salida está dividida en secciones. La primera sección muestra información sobre la versión de `dig` instalada y la consulta enviada, junto con las opciones utilizadas para el comando. A continuación muestra información sobre la consulta y la respuesta.

La siguiente sección muestra información sobre las extensiones EDNS utilizadas y la consulta. En el ejemplo, se utiliza la extensión cookie. `dig` está buscando un registro A para `learning.lpi.org`.

La siguiente sección muestra el resultado de la consulta. El número de la segunda columna es el TTL del recurso en segundos.

El resto de la salida proporciona información sobre los servidores de nombres del dominio, incluyendo los registros NS del servidor junto con los registros A y AAAA de los servidores en el registro NS del dominio.

Al igual que `host`, puede especificar un tipo de registro con la opción `-t`:

```
$ dig -t SOA lpi.org

; <<>> DiG 9.11.5-P4-5.1+deb10u1-Debian <<>> -t SOA lpi.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16695
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 185c67140a63baf46c4493215ef8906f7bfbe15bdca3b01a (good)
;; QUESTION SECTION:
;lpi.org.           IN  SOA

;; ANSWER SECTION:
lpi.org.        600 IN  SOA dns1.easydns.com. zone.easydns.com. 1593109612 3600 600
1209600 300

;; AUTHORITY SECTION:
lpi.org.        81989   IN  NS  dns1.easydns.com.
lpi.org.        81989   IN  NS  dns2.easydns.net.
lpi.org.        81989   IN  NS  dns3.easydns.ca.

;; ADDITIONAL SECTION:
dns1.easydns.com.   168271  IN  A   64.68.192.10
dns2.easydns.net.   165815  IN  A   198.41.222.254
dns3.easydns.ca.    107 IN  A   64.68.196.10
dns1.easydns.com.   168271  IN  AAAA    2400:cb00:2049:1::a29f:1835
dns2.easydns.net.   165815  IN  AAAA    2400:cb00:2049:1::c629:defe

;; Query time: 94 msec
;; SERVER: 192.168.1.20#53(192.168.1.20)
;; WHEN: Sun Jun 28 08:43:27 EDT 2020
;; MSG SIZE  rcvd: 298
```

El comando dig tiene muchas opciones para afinar tanto la salida como la consulta enviada al servidor. Estas opciones comienzan con `+`. Una de ellas es la opción `short`, que suprime toda la salida

excepto el resultado:

```
$ dig +short lpi.org
65.39.134.165
$ dig +short -t SOA lpi.org
dns1.easydns.com. zone.easydns.com. 1593109612 3600 600 1209600 300
```

Este es un ejemplo de cómo desactivar la extensión de la cookie EDNS:

```
$ dig +nocookie -t MX lpi.org

; <<>> DiG 9.11.5-P4-5.1+deb10u1-Debian <<>> +nocookie -t MX lpi.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47774
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 3, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;lpi.org.           IN  MX

;; ANSWER SECTION:
lpi.org.        468 IN  MX  0 aspmx.l.google.com.
lpi.org.        468 IN  MX  10 aspmx4.googlemail.com.
lpi.org.        468 IN  MX  10 aspmx5.googlemail.com.
lpi.org.        468 IN  MX  10 aspmx2.googlemail.com.
lpi.org.        468 IN  MX  10 aspmx3.googlemail.com.
lpi.org.        468 IN  MX  5 alt2.aspmx.l.google.com.
lpi.org.        468 IN  MX  5 alt1.aspmx.l.google.com.

;; AUTHORITY SECTION:
lpi.org.        77130   IN  NS  dns2.easydns.net.
lpi.org.        77130   IN  NS  dns3.easydns.ca.
lpi.org.        77130   IN  NS  dns1.easydns.com.

;; ADDITIONAL SECTION:
dns1.easydns.com.   76140   IN  A   64.68.192.10
dns2.easydns.net.   73684   IN  A   198.41.222.254
dns1.easydns.com.   76140   IN  AAAA    2400:cb00:2049:1::a29f:1835
dns2.easydns.net.   73684   IN  AAAA    2400:cb00:2049:1::c629:defe

;; Query time: 2 msec
;; SERVER: 192.168.1.20#53(192.168.1.20)
;; WHEN: Mon Jun 29 10:18:58 EDT 2020
;; MSG SIZE  rcvd: 389
```

Ejercicios guiados

1. ¿Qué hará el siguiente comando?

```
getent group openldap
```

2. ¿Cuál es la mayor diferencia entre `getent` y las otras herramientas cubiertas, `host` y `dig`?

3. ¿Qué opción de `dig` y `host` se utiliza para especificar el tipo de registro que se desea recuperar?

4. ¿Cuál de las siguientes es una entrada correcta de `/etc/hosts`?

`::1 localhost`	
`localhost 127.0.0.1`	

5. ¿Qué opción de `getent` se utiliza para especificar qué fuente de datos debe utilizarse para realizar una búsqueda?

Ejercicios de exploración

1. Si usted editara el `/etc/resolv.conf` que aparece a continuación con un editor de texto, ¿qué es probable que ocurra?

```
# Generated by NetworkManager
nameserver 192.168.1.20
```

Los cambios serán sobrescritos por NetworkManager.	
NetworkManager actualizará su configuración con sus cambios.	
Sus cambios no afectarán al sistema.	
NetworkManager se desactivará.	

2. ¿Qué significa la siguiente línea en `/etc/nsswitch.conf`?

```
hosts: files [SUCCESS=continue] dns
```

3. Teniendo en cuenta el siguiente `/etc/resolv.conf` ¿por qué el sistema no está resolviendo los nombres a través de DNS?

```
search lpi.org
#nameserver fd00:ffff::1:53
#nameserver 10.0.1.53
```

4. ¿Qué hace el comando `dig +noall +answer +question lpi.org`?

5. ¿Cómo puede anular los valores predeterminados de `dig` sin especificarlos en la línea de comandos?

Resumen

El comando `getent` es una gran herramienta para ver los resultados de las llamadas a resolver para consultas DNS simples, `host` es fácil de usar y produce una salida directa. Si necesitas información detallada o necesitas afinar una consulta DNS, `dig` es probablemente tu mejor opción.

Gracias a la posibilidad de añadir plugins de bibliotecas compartidas y configurar el comportamiento de resolver, Linux tiene un excelente soporte para la resolución de nombres y números de varios tipos. El programa `getent` puede utilizarse para resolver nombres utilizando las bibliotecas de resolución. Los programas `host` y `dig` se pueden utilizar para consultar servidores DNS.

El archivo `/etc/nsswitch.conf` se utiliza para configurar el comportamiento de resolver. Puede cambiar las fuentes de datos y añadir alguna lógica condicional simple para los tipos de nombre con múltiples fuentes.

El DNS se configura editando `/etc/resolv.conf`. Muchas distribuciones tienen herramientas que gestionan este archivo por usted, así que asegúrese de revisar la documentación de su sistema si los cambios manuales no persisten.

El archivo `/etc/hosts` se utiliza para resolver nombres de host a IPs y viceversa. Normalmente se utiliza para definir nombres, como `localhost` que no están disponibles a través de DNS.

Es posible dejar comentarios en los archivos de configuración tratados en esta lección. Cualquier texto a la derecha de `#` es ignorado por el sistema.

Respuesta a los ejercicios guiados

1. ¿Qué hará el siguiente comando?

```
getent group openldap
```

Leerá `/etc/nsswitch.conf`, buscará el grupo `openldap` de las fuentes listadas y mostrará información sobre él si lo encuentra.

2. ¿Cuál es la mayor diferencia entre `getent` y las otras herramientas cubiertas, `host` y `dig`?

`getent` busca nombres usando las librerías de resolución, los otros sólo consultan el DNS. `getent` se puede utilizar para solucionar los problemas de su `/etc/nsswitch.conf` y la configuración de las bibliotecas de resolución de nombres que su sistema está configurado para utilizar. `host` y `dig` se utilizan para buscar registros DNS.

3. ¿Qué opción de `dig` y `host` se utiliza para especificar el tipo de registro que se desea recuperar?

Ambos programas utilizan `-t` para especificar el tipo de registro que se desea buscar.

4. ¿Cuál de las siguientes es una entrada correcta de `/etc/hosts`?

`::1 localhost`	X
`localhost 127.0.0.1`	

`::1 localhost` es la línea correcta. La columna de la izquierda es siempre una dirección IPv4 o IPv6.

5. ¿Qué opción de `getent` se utiliza para especificar qué fuente de datos debe utilizarse para realizar una búsqueda?

La opción `-s` se utiliza para especificar la fuente de datos. Por ejemplo:

```
$ getent -s files hosts learning.lpi.org
192.168.10.25   learning.lpi.org
$ getent -s dns hosts learning.lpi.org
208.94.166.198  learning.lpi.org
```

Respuestas a los ejercicios de exploración

1. Si usted editara el `/etc/resolv.conf` que aparece a continuación con un editor de texto, ¿qué es probable que ocurra?

```
# Generated by NetworkManager
nameserver 192.168.1.20
```

Los cambios serán sobrescritos por NetworkManager.	
NetworkManager actualizará su configuración con sus cambios.	
Sus cambios no afectarán al sistema.	
NetworkManager se desactivará.	

2. ¿Qué significa la siguiente línea en `/etc/nsswitch.conf`?

```
hosts: files [SUCCESS=continue] dns
```

Las búsquedas de nombres de host comprobarán primero sus archivos `/etc/hosts` y luego el DNS. Si se encuentra una entrada en los archivos y en el DNS, se utilizará la entrada en el DNS.

3. Teniendo en cuenta el siguiente `/etc/resolv.conf` ¿por qué el sistema no está resolviendo los nombres a través de DNS?

```
search lpi.org
#nameserver fd00:ffff::1:53
#nameserver 10.0.1.53
```

Ambos servidores DNS están comentados y no hay ningún servidor DNS funcionando en el host local.

4. ¿Qué hace el comando `dig +noall +answer +question lpi.org`?

Busca el registro A de `lpi.org` y muestra sólo la consulta y la respuesta.

5. ¿Cómo puede anular los valores predeterminados de `dig` sin especificarlos en la línea de comandos?

Creas un archivo `.digrc` en tu directorio personal.



Tema 110: Seguridad



110.1 Tareas de administración de seguridad

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 110.1

Importancia

3

Áreas de conocimiento clave

- Auditar un sistema para encontrar archivos con el bit suid/sgid activo.
- Establecer o cambiar contraseñas de usuario e información de caducidad de contraseña.
- Saber usar nmap y netstat para descubrir puertos abiertos en un sistema.
- Establecer límites en los inicios de sesión de los usuarios, en los procesos y en el uso de memoria.
- Determinar que usuarios han iniciado sesión en el sistema o están actualmente conectados.
- Configuración y uso básicos de sudo.

Lista parcial de archivos, términos y utilidades

- `find`
- `passwd`
- `fuser`
- `lsof`
- `nmap`
- `chage`
- `netstat`

- `sudo`
- `/etc/sudoers`
- `su`
- `usermod`
- `ulimit`
- `who`, `w`, `last`



110.1 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	110 Seguridad
Objetivo:	110.1 Tareas de administración de seguridad
Lección:	1 de 1

Introducción

La seguridad es una necesidad en la administración de sistemas. Como un buen administrador de sistemas Linux debe vigilar una serie de cosas como los permisos especiales en los archivos, el vencimiento de las contraseñas de los usuarios, los puertos y sockets abiertos, la limitación del uso de los recursos del sistema, el manejo de los usuarios con sesión iniciada y la escalación de privilegios a través de `su` y `sudo`. En esta lección revisaremos cada uno de estos temas. === Comprobación de archivos con el conjunto SUID y SGID

Además del conjunto tradicional de permisos de *lectura*, *escritura* y *ejecución*, los archivos en un sistema Linux pueden tener también algunos permisos especiales como los bits *SUID* o *SGID*.

El bit SUID permite que el fichero se ejecute con los privilegios del propietario del archivo. Se representa numéricamente por `4000` y simbólicamente por `s` o `S` en el bit de permiso *execute* del propietario. Un ejemplo clásico de un archivo ejecutable con el permiso SUID establecido es `passwd`:

```
carol@debian:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 63736 jul 27  2018 /usr/bin/passwd
```

La `s` minúscula en `rws` indica la presencia del SUID en el archivo, junto con el permiso de ejecución. Una `S` mayúscula en su lugar (`rwS`) significaría que el permiso *execute* subyacente no está establecido.

> **NOTE**
>
> Aprenderá sobre `passwd` en la siguiente sección. La utilidad es empleada principalmente por `root` para establecer/cambiar las contraseñas de los usuarios (por ejemplo: `passwd carol`). Sin embargo, los usuarios normales también pueden utilizarla para cambiar sus propias contraseñas. Por lo tanto, viene con el conjunto SUID.

Por otro lado, el bit SGID puede establecerse tanto en archivos como en directorios. En el caso de los archivos, su comportamiento es equivalente al de SUID, pero los privilegios son los del propietario del grupo. Sin embargo, cuando se establece en un directorio, permitirá que todos los archivos creados en él hereden la propiedad del grupo del directorio. Al igual que SUID, SGID se representa simbolicamente por `s` o `S` en el bit de permiso *execute* del grupo. Numéricamente, se representa por `2000`. Puede establecer el SGID en un directorio utilizando `chmod`. Es necesario `2` (SGID) a los permisos tradicionales (`755` en nuestro caso):

```
carol@debian:~$ ls -ld shared_directory
drwxr-xr-x 2 carol carol 4096 may 30 23:55 shared_directory
carol@debian:~$ sudo chmod 2755 shared_directory/
carol@debian:~$ ls -ld shared_directory
drwxr-sr-x 2 carol carol 4096 may 30 23:55 shared_directory
```

Para buscar archivos con uno o ambos conjuntos de SUID y SGID puede utilizar el comando `find` y hacer uso de la opción `-perm`. Puede utilizar tanto valores numéricos como simbólicos. Los valores — a su vez — pueden ser pasados solos o precedidos por un guión (`-`) o una barra inclinada (`/`). El significado es el siguiente:

`-perm` `numeric-value` **o** **`-perm`** `symbolic-value`
: Encontrar archivos que tengan el permiso especial *exclusively*

`-perm` `-numeric-value` **o** **`-perm`** `-symbolic-value`
: Buscar archivos que tengan el permiso especial y otros permisos

`-perm /``numeric-value` **o** **`-perm /`**`symbolic-value`
: Encontrar archivos que tengan alguno de los permisos especiales (y otros permisos)

Por ejemplo, para buscar archivos con *sólo* SUID en el directorio de trabajo actual, se utilizará el siguiente comando:


```
carol@debian:~$ find . -perm 4000
carol@debian:~$ touch file
carol@debian:~$ chmod 4000 file
carol@debian:~$ find . -perm 4000
./file
```

Observe como — al no haber ningún archivo que tenga exclusivamente el SUID — creamos uno para mostrar alguna salida. Puede ejecutar el mismo comando en notación simbólica:

```
carol@debian:~$ find . -perm u+s
./file
```

Para buscar archivos que coincidan con el SUID (independientemente de cualquier otro permiso) en el directorio `/usr/bin/`, puede utilizar cualquiera de los siguientes comandos:

```
carol@debian:~$ sudo find /usr/bin -perm -4000
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/su
carol@debian:~$ sudo find /usr/bin -perm -u+s
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/su
```

Si busca archivos en el mismo directorio con el bit SGID activado, puede ejecutar `find /usr/bin/ -perm -2000` o `find /usr/bin/ -perm -g+s`.

Por último, para buscar archivos con cualquiera de los dos permisos especiales establecidos, añada `4` y `2` y utilice `/`:

```
carol@debian:~$ sudo find /usr/bin -perm /6000
/usr/bin/dotlock.mailutils
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/wall
/usr/bin/ssh-agent
/usr/bin/chage
/usr/bin/dotlockfile
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/expiry
/usr/bin/sudo
/usr/bin/bsd-write
/usr/bin/crontab
/usr/bin/su
```

Gestión y caducidad de contraseñas

Como se ha indicado anteriormente, puede emplear la utilidad `passwd` para cambiar su propia contraseña como usuario normal. Además, puede pasar la opción `-S` o `--status` para obtener información sobre el estado de su cuenta:

```
carol@debian:~$ passwd -S
carol P 12/07/2019 0 99999 7 -1
```

A continuación se desglosan los siete campos que se obtienen en la salida:

`carol`
: Nombre de acceso del usuario.

`P`
: Indica que el usuario tiene una contraseña válida (`P`); otros valores posibles son `L` para una contraseña bloqueada y `NP` para ninguna contraseña.

`12/07/2019`
: Fecha del último cambio de contraseña.

`0`

Edad mínima en días (el número mínimo de días entre cambios de contraseña). Un valor de `0` significa que la contraseña puede cambiarse en cualquier momento.

`99999`

Edad máxima en días (el número máximo de días que la contraseña es válida). Un valor de `99999` desactivará la caducidad de la contraseña.

`7`

Período de advertencia en días (el número de días antes de la expiración de la contraseña que un usuario será advertido).

`-1`

Periodo de inactividad de la contraseña en días (el número de días inactivos después de la expiración de la contraseña antes de que la cuenta se bloquee). Un valor de `-1` eliminará la inactividad de una cuenta.

Aparte de informar sobre el estado de las cuentas, se puede utilizar el comando `passwd` como root para llevar a cabo algunas tareas básicas de mantenimiento de cuentas. Puede bloquear y desbloquear cuentas, forzar a un usuario a cambiar su contraseña en el siguiente inicio de sesión y eliminar la contraseña de un usuario con las opciones `-l`, `-u`, `-e` y `-d`, respectivamente.

Para probar estas opciones es conveniente introducir el comando `su` en este punto. A través de `su` se puede cambiar de usuario durante una sesión de inicio de sesión. Por ejemplo, usemos `passwd` como root para bloquear la contraseña de `carol`. Entonces cambiaremos a `carol` y comprobaremos el estado de nuestra cuenta para verificar que la contraseña ha sido --de hecho-- bloqueada (`L`) y no puede ser cambiada. Finalmente, volviendo al usuario root, desbloquearemos la contraseña de `carol`:

```
root@debian:~# passwd -l carol
passwd: password expiry information changed.
root@debian:~# su - carol
carol@debian:~$ passwd -S
carol L 05/31/2020 0 99999 7 -1
carol@debian:~$ passwd
Changing password for carol.
Current password:
passwd: Authentication token manipulation error
passwd: password unchanged
carol@debian:~$ exit
logout
root@debian:~# passwd -u carol
passwd: password expiry information changed.
```

También puede bloquear y desbloquear la contraseña de un usuario con el comando `usermod`:

Bloquear la contraseña del usuario `carol`

`usermod -L carol` o `usermod --lock carol`.

Desbloquear la contraseña del usuario `carol`

`usermod -U carol` o `usermod --unlock carol`.

NOTE Las opciones `-f` o `--inactive`, `usermod` también pueden utilizarse para establecer el número de días antes de que se desactive una cuenta con una contraseña expirada (por ejemplo: `usermod -f 3 carol`).

Además de `passwd` y `usermod`, el comando más directo para tratar la caducidad de contraseñas y cuentas es `chage` ("change age"). Como root, puede pasarle a `chage` la opción `-l` (o `--list`) seguido de un nombre de usuario para que se imprima en la pantalla la contraseña actual de ese usuario y la información de caducidad de la cuenta; como usuario normal, puede ver su propia información:

```
carol@debian:~$ chage -l carol
Last password change                                    : Aug 06, 2019
Password expires                                : never
Password inactive                               : never
Account expires                                 : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Ejecutado sin opciones y sólo seguido de un nombre de usuario, `chage` se comportará de forma interactiva:

```
root@debian:~# chage carol
Changing the aging information for carol
Enter the new value, or press ENTER for the default

    Minimum Password Age [0]:
    Maximum Password Age [99999]:
    Last Password Change (YYYY-MM-DD) [2020-06-01]:
    Password Expiration Warning [7]:
    Password Inactive [-1]:
    Account Expiration Date (YYYY-MM-DD) [-1]:
```

Las opciones para modificar los diferentes ajustes de `chage` son las siguientes:

`-m` `days username` **o** **`--mindays`** `days username`

Especifica el número mínimo de días entre cambios de contraseña (por ejemplo: `chage -m 5 carol`). Un valor de `0` permitirá al usuario cambiar su contraseña en cualquier momento.

`-M` `days username` **o** **`--maxdays`** `days username`

Especifica el número máximo de días que la contraseña será válida (por ejemplo: `chage -M 30 carol`). Para desactivar la caducidad de la contraseña, es habitual dar a esta opción un valor de `99999`.

`-d` `days username` **o** **`--lastday`** `days username`

Especifica el número de días desde que la contraseña fue cambiada por última vez (por ejemplo: `chage -d 10 carol`). Un valor de `0` obligará al usuario a cambiar su contraseña en el siguiente inicio de sesión.

`-W` `days username` **o** **`--warndays`** `days username`

Especifica el número de días que se le recordará al usuario que su contraseña ha caducado.

`-I` `days username` **o** **`--inactive`** `days username`

Especifica el número de días inactivos después de la expiración de la contraseña (por ejemplo: `chage -I 10 carol`)—lo mismo que `usermod -f` o `usermod --inactive`. Una vez que haya pasado ese número de días, la cuenta se bloqueará. Sin embargo, con un valor de `0`, la cuenta no se bloqueará.

-E `date username` **o** **--expiredate** `date username`

Especifica la fecha (o el número de días desde *la época*—el 1 de enero de 1970) en la que se bloqueará la cuenta. Normalmente se expresa en el formato `YYYY-MM-DD` (por ejemplo: `chage -E 2050-12-13 carol`).

> **NOTE** Puede encontrar más información sobre `passwd`, `usermod` y `chage` y sus opciones consultando sus respectivas páginas de manual.

Descubrir los puertos abiertos

Cuando se trata de vigilar los puertos abiertos, hay cuatro potentes utilidades presentes en la mayoría de los sistemas Linux: `lsof`, `fuser`, `netstat` y `nmap`. Las cubriremos en esta sección.

`lsof` significa "listar archivos abiertos", lo cual no es poca cosa teniendo en cuenta que—para Linux—todo es un archivo. De hecho, si escribe `lsof` en el terminal, obtendrá un gran listado de archivos regulares, archivos de dispositivos, sockets, etc. Sin embargo, por el bien de esta lección, nos centraremos principalmente en los puertos. Para imprimir el listado de todos los archivos de red de "Internet", ejecute `lsof` con la opción `-i`:

```
root@debian:~# lsof -i
COMMAND  PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
dhclient 357     root    7u  IPv4  13493      0t0  UDP *:bootpc
sshd     389     root    3u  IPv4  13689      0t0  TCP *:ssh (LISTEN)
sshd     389     root    4u  IPv6  13700      0t0  TCP *:ssh (LISTEN)
apache2  399     root    3u  IPv6  13826      0t0  TCP *:http (LISTEN)
apache2  401 www-data    3u  IPv6  13826      0t0  TCP *:http (LISTEN)
apache2  402 www-data    3u  IPv6  13826      0t0  TCP *:http (LISTEN)
sshd     557     root    3u  IPv4  14701      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:60510 (ESTABLISHED)
sshd     569    carol    3u  IPv4  14701      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:60510 (ESTABLISHED)
```

Aparte del servicio `bootpc`—que es utilizado por DHCP—la salida muestra dos servicios que están escuchando conexiones—`ssh` y el servidor web Apache (`http`)—así como dos conexiones SSH establecidas. Puede especificar un host en particular con la notación `@ip-address` para comprobar sus conexiones:


```
root@debian:~# lsof -i@192.168.1.7
COMMAND PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd    557  root    3u  IPv4  14701      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:60510 (ESTABLISHED)
sshd    569 carol    3u  IPv4  14701      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:60510 (ESTABLISHED)
```

NOTE Para imprimir sólo los archivos de red IPv4 e IPv6, utilice las opciones `-i4` y `-i6` respectivamente.

Asimismo, puede filtrar por puerto pasando la opción `-i` (o `-i@ip-address`) al argumento `:port`.

```
root@debian:~# lsof -i :22
COMMAND PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd    389  root    3u  IPv4  13689      0t0  TCP *:ssh (LISTEN)
sshd    389  root    4u  IPv6  13700      0t0  TCP *:ssh (LISTEN)
sshd    557  root    3u  IPv4  14701      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:60510 (ESTABLISHED)
sshd    569 carol    3u  IPv4  14701      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:60510 (ESTABLISHED)
```

Los puertos múltiples se separan con comas (y los rangos se especifican con un guión):

```
root@debian:~# lsof -i@192.168.1.7:22,80
COMMAND PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd    705  root    3u  IPv4  13960      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:44766 (ESTABLISHED)
sshd    718 carol    3u  IPv4  13960      0t0  TCP 192.168.1.7:ssh-
>192.168.1.4:44766 (ESTABLISHED)
```

NOTE La cantidad de opciones de que dispone `lsof` es bastante impresionante. Para saber más, consulte su página de manual.

El siguiente en la lista de comandos de red es `fuser`. Su propósito principal es encontrar el "usuario de un fichero", lo que implica saber qué procesos están accediendo a qué ficheros; también da alguna otra información como el tipo de acceso. Por ejemplo, para comprobar el directorio de trabajo actual, basta con ejecutar `fuser`. Sin embargo, para obtener un poco más de información, es conveniente utilizar la opción verbose (`-v` o `--verbose`):

```
root@debian:~# fuser .
/root:                 580c
root@debian:~# fuser -v .
                     USER        PID ACCESS COMMAND
/root:               root        580 ..c.. bash
```

Desglosemos la salida:

`File`
: El archivo del que estamos obteniendo información (`/root`).

`USER`
: El propietario del fichero (`root`).

`PID`
: El identificador del proceso (`580`).

`ACCESS`
: Tipo de acceso (`..c..`). Uno de:

  `c`
  : Directorio actual.

  `e`
  : Ejecutables que se llevan a cabo.

  `f`
  : Abrir archivo (se omite en el modo de visualización por defecto).

  `F`
  : Abrir archivo para escribir (se omite en el modo de visualización por defecto).

  `r`
  : Directorio raíz.

  `m`
  : archivo mmap'ed o biblioteca compartida.

`.`
: Marcador de posición (omitido en el modo de visualización por defecto).

`COMMAND`
: El comando afiliado al archivo (`bash`).

Con la opción `-n` (o `--namespace`), puede encontrar información sobre los puertos/sockets de red. También debe proporcionar el protocolo de red y el número de puerto. Así para obtener información sobre el servidor web Apache ejecutará el siguiente comando:

```
root@debian:~# fuser -vn tcp 80
                     USER        PID ACCESS COMMAND
80/tcp:              root        402 F.... apache2
                     www-data    404 F.... apache2
                     www-data    405 F.... apache2
```

> **NOTE** `fuser` también se puede utilizar para matar los procesos que acceden al archivo con las opciones `-k` o `--kill` (por ejemplo: `fuser -k 80/tcp`). Consulte la página del manual para obtener información más detallada.

Pasemos ahora a `netstat`. Esta es una herramienta de red muy versátil que se utiliza principalmente para imprimir "estadísticas de red".

Ejecutado sin opciones, `netstat` mostrará tanto las conexiones activas a Internet como los sockets de Unix. Debido al tamaño del listado, es posible que quiera canalizar su salida a través de `less`:

```
carol@debian:~$ netstat |less
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.7:ssh         192.168.1.4:55444       ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    10509
/run/systemd/journal/syslog
unix  3      [ ]         DGRAM                    10123    /run/systemd/notify
(...)
```

Para listar sólo los puertos y sockets de "escucha", se utilizarán las opciones `-l` o `--listening`. Las opciones `-t`/`--tcp` y `-u`/`--udp` pueden añadirse para filtrar por protocolo TCP y UDP, respectivamente (también pueden combinarse en el mismo comando). Asimismo, `-e`/`--extend`

mostrará información adicional:

```
carol@debian:~$ netstat -lu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
carol@debian:~$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:http               [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 localhost:smtp          [::]:*                  LISTEN
carol@debian:~$ netstat -lute
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User       Inode
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
root       13729
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
root       14372
tcp6       0      0 [::]:http               [::]:*                  LISTEN
root       14159
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
root       13740
tcp6       0      0 localhost:smtp          [::]:*                  LISTEN
root       14374
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
root       13604
```

Si omite la opción `-l`, sólo se mostrarán las conexiones establecidas:

```
carol@debian:~$ netstat -ute
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User       Inode
tcp        0      0 192.168.1.7:ssh         192.168.1.4:39144       ESTABLISHED
root       15103
```

Si sólo le interesa la información numérica relativa a los puertos y hosts, puede utilizar la opción `-n` o `--numeric` para imprimir sólo los números de puerto y las direcciones IP. Observe cómo `ssh` se

convierte en `22` al añadir `-n` al comando anterior:

```
carol@debian:~$ netstat -uten
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User       Inode
tcp        0      0 192.168.1.7:22          192.168.1.4:39144       ESTABLISHED 0
15103
```

Como puede ver, es posible utilizar comandos `netstat` muy útiles y productivos combinando algunas de sus opciones. Navegue por las páginas del manual para aprender más y encontrar las combinaciones que mejor se adapten a sus necesidades.

Por último, presentaremos `nmap` — o el “network mapper”. Otra utilidad muy potente, este escáner de puertos se ejecuta especificando una dirección IP o un nombre de host:

```
root@debian:~# nmap localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2020-06-04 19:29 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
```

Aparte de un solo host, `nmap` le permite escanear:

Múltiples hosts

Separándolos con espacios (por ejemplo: `nmap localhost 192.168.1.7`).

Rangos de hosts

Utilizando un guión (por ejemplo: `nmap 192.168.1.3-20`).

Subredes

Utilizando un comodín o una notación CIDR (por ejemplo: `nmap 192.168.1.*` o `nmap 192.168.1.0/24`). Puede excluir determinados hosts (por ejemplo: `nmap 192.168.1.0/24 --exclude 192.168.1.7`).

Para escanear un puerto concreto, utilice la opción `-p` seguida del número de puerto o del nombre del servicio (`nmap -p 22` y `nmap -p ssh` le darán la misma salida):

```
root@debian:~# nmap -p 22 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2020-06-04 19:54 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000024s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

También puede escanear varios puertos o rangos de puertos utilizando comas y guiones, respectivamente:

```
root@debian:~# nmap -p ssh,80 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2020-06-04 19:58 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000051s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

```
root@debian:~# nmap -p 22-80 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2020-06-04 19:58 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000011s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 57 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.47 seconds
```

Otras dos opciones importantes y útiles de `nmap` son:

`-F`
: Ejecuta un escaneo rápido en los 100 puertos más comunes.

`-v`
: Obtiene una salida más detallada (`-vv` imprimirá una salida con más información).

> **NOTE** `nmap` puede ejecutar comandos bastante complejos haciendo uso de los tipos de exploración. Sin embargo, ese tema está fuera del alcance de esta lección.

Límites en los inicios de sesión de los usuarios, los procesos y el uso de la memoria

Los recursos en un sistema Linux no son ilimitados, por lo que — como administrador del sistema — debe asegurar un buen equilibrio entre los límites de los usuarios sobre los recursos y el correcto funcionamiento del sistema operativo. `ulimit` puede ayudarle en este sentido.

`ulimit` se ocupa de los límites *soft* y *hard* — especificados por las opciones `-S` y `-H`, respectivamente. Si se ejecuta sin opciones ni argumentos, `ulimit` mostrará los bloques de archivos con límites flexibles del usuario actual:

```
carol@debian:~$ ulimit
unlimited
```

Con la opción `-a`, `ulimit` mostrará todos los límites flexibles actuales (lo mismo que `-Sa`); para mostrar todos los límites estrictos actuales, utilice `-Ha`:

```
carol@debian:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
(...)
carol@debian:~$ ulimit -Ha
core file size          (blocks, -c) unlimited
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
(...)
```

Los recursos del shell disponibles se especifican mediante opciones como:

`-b`
: tamaño máximo del búfer en el socket

`-f`
: tamaño máximo de los archivos escritos por el shell y sus hijos

`-l`
: tamaño máximo que se puede bloquear en la memoria

`-m`
: tamaño máximo del conjunto residente (RSS) — la porción actual de memoria que tiene un proceso en la memoria principal (RAM)

`-v`
: cantidad máxima de memoria virtual

`-u`
: número máximo de procesos disponibles para un solo usuario

Así, para mostrar los límites se utilizará `ulimit` seguido de `-S` (flexible) o `-H`(estricto) y la opción de recurso; si no se suministra ni `-S` ni `-H`, se mostrarán los límites flexibles:

```
carol@debian:~$ ulimit -u
10000
carol@debian:~$ ulimit -Su
10000
carol@debian:~$ ulimit -Hu
15672
```

Del mismo modo, para establecer nuevos límites en un recurso concreto se especificará `S` o `H`, seguido de la opción de recurso correspondiente y el nuevo valor. Este valor puede ser un número o las palabras especiales `soft` (límite flexible actual), `hard` (límite estricto actual) o `unlimited` (sin límite). Si no se especifica ni `S` ni `H`, se establecerán ambos límites. Por ejemplo, leamos primero el valor del tamaño máximo actual de los archivos escritos por el shell y sus hijos:

```
root@debian:~# ulimit -Sf
unlimited
root@debian:~# ulimit -Hf
unlimited
```

Ahora, cambiemos el valor de `ilimitado` a `500` bloques sin especificar ni `-S` ni `-H`. Observe cómo se modifican tanto los límites flexibles como los estrictos:

```
root@debian:~# ulimit -f 500
root@debian:~# ulimit -Sf
500
root@debian:~# ulimit -Hf
500
```

Por último, disminuiremos sólo el límite flexible a `200` bloques:

```
root@debian:~# ulimit -Sf 200
root@debian:~# ulimit -Sf
200
root@debian:~# ulimit -Hf
500
```

Los límites estrictos sólo pueden ser aumentados por el usuario root. Por otro lado, los usuarios regulares pueden disminuir los límites estrictos y aumentar los límites flexibles hasta el valor de los límites duros. Para hacer que los nuevos valores de los límites sean persistentes a través de los reinicios, debe escribirlos en el archivo `/etc/security/limits.conf`. Este es también el archivo utilizado por el administrador para aplicar restricciones a determinados usuarios.

NOTE Se advierte que no hay una página man de `ulimit` como tal. Es una integración de bash, así que tiene que consultar la página man de bash para aprender sobre este.

Tratar con usuarios registrados

Otro de los trabajos como administrador de sistemas implica llevar un registro de los usuarios conectados. Hay tres utilidades que pueden ayudar con esas tareas: `last`, `who` y `w`.

`last` imprime un listado de los últimos usuarios conectados con la información más reciente en la parte superior:

```
root@debian:~# last
carol    pts/0        192.168.1.4      Sat Jun  6 14:25   still logged in
reboot   system boot  4.19.0-9-amd64   Sat Jun  6 14:24   still running
mimi     pts/0        192.168.1.4      Sat Jun  6 12:07 - 14:24  (02:16)
reboot   system boot  4.19.0-9-amd64   Sat Jun  6 12:07 - 14:24  (02:17)
(...)
wtmp begins Sun May 31 14:14:58 2020
```

Considerando el listado truncado, obtenemos información sobre los dos últimos usuarios del sistema. Las dos primeras líneas nos hablan del usuario `carol`; las dos siguientes, del usuario `mimi`. La información es la siguiente:

1. El usuario `carol` en la terminal `pts/0` desde el host `192.168.1.4` inició su sesión el `sábado 6 de junio` a las `14:25` y todavía está `conectada`. El sistema, que utiliza el kernel `4.19.0-9-amd64`, se inició (`reboot system boot`) el `sábado 6 de junio` a las `14:24` y sigue funcionando.
2. El usuario `mimi` en la terminal `pts/0` desde el host `192.168.1.4` inició su sesión el `sábado 6 de junio` a las `12:07` y cerró la sesión a las `14:24` (la sesión duró un total de `(02:16)` horas). El sistema que utiliza el kernel `4.19.0-9-amd64`, se inició (`reboot system boot`) el `sábado 6 de junio` a las `12:07` y se apagó a las `14:24` (estuvo funcionando durante `(02:17)` horas).

NOTE La línea `wtmp begins Sun May 31 14:14:58 2020` se refiere a `/var/log/wtmp`, que es el archivo de registro especial del que `last` obtiene la información.

Puede pasarle a `last` un nombre de usuario para que sólo se muestren las entradas de ese usuario:

```
root@debian:~# last carol
carol    pts/0        192.168.1.4      Sat Jun  6 14:25   still logged in
carol    pts/0        192.168.1.4      Sat Jun  6 12:07 - 14:24  (02:16)
carol    pts/0        192.168.1.4      Fri Jun  5 00:48 - 01:28  (00:39)
(...)
```

En cuanto a la segunda columna (terminal), `pts` significa *Pseudo Terminal Slave* - en contraposición a un terminal *TeleTYpewriter* o `tty` propiamente dicho; `0` se refiere al primero (la cuenta comienza en cero).

NOTE Para comprobar los intentos fallidos de inicio de sesión, ejecute `lastb` en lugar de `last`.

Las utilidades `who` y `w` se centran en los usuarios actualmente conectados y son bastante similares. La

primera muestra quién está conectado, mientras que la segunda también muestra información sobre lo que están haciendo.

Cuando se ejecuta sin opciones, `who` mostrará cuatro columnas correspondientes al usuario conectado, el terminal, la fecha, la hora, y el nombre del host:

```
root@debian:~# who
carol    pts/0        2020-06-06 17:16 (192.168.1.4)
mimi     pts/1        2020-06-06 17:28 (192.168.1.4)
```

`who` acepta una serie de opciones, entre las que podemos destacar las siguientes:

`-b`,`--boot`
: Muestra la hora del último arranque del sistema.

`-r`,`--runlevel`
: Muestra el nivel de ejecución actual.

`-H`,`--heading`
: Imprime los títulos de las columnas.

En comparación con `who`, `w` da una salida un poco más detallada:

```
root@debian:~# w
 17:56:12 up 40 min,  2 users,  load average: 0.04, 0.12, 0.09
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
carol    pts/0    192.168.1.4      17:16    1.00s  0.15s  0.05s sshd: carol [priv]
mimi     pts/1    192.168.1.4      17:28   15:08   0.05s  0.05s -bash
```

La línea superior ofrece información sobre la hora actual (`17:56:12`), el tiempo que lleva el sistema en funcionamiento (`up 40 min`), el número de usuarios conectados en ese momento (`2 usuarios`) y los números de la media de carga (`media de carga: 0,04, 0,12, 0,09`). Estos valores se refieren al número de trabajos en la cola de ejecución promediados en los últimos 1, 5 y 15 minutos, respectivamente.

A continuación, encontrará ocho columnas; vamos a desglosarlas:

`USER`
: Nombre de inicio de sesión del usuario.

TTY
: Nombre del terminal en el que se encuentra el usuario.

FROM
: Host remoto desde el que el usuario se ha conectado.

LOGIN@
: Hora de inicio de sesión.

IDLE
: Tiempo de inactividad.

JCPU
: Tiempo utilizado por todos los procesos conectados a la tty (incluidos los trabajos en segundo plano que se están ejecutando actualmente).

PCPU
: Tiempo utilizado por el proceso actual (el que se muestra bajo `WHAT`).

WHAT
: Línea de comandos del proceso actual.

Al igual que con `who` puede pasar nombres de usuario `w`:

```
root@debian:~# w mimi
 18:23:15 up  1:07,  2 users,  load average: 0.00, 0.02, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
mimi     pts/1    192.168.1.4      17:28    9:23   0.06s  0.06s -bash
```

Configuración y uso básico de `sudo`

Como ya se ha señalado en esta lección, `su` permite cambiar a cualquier otro usuario del sistema siempre que se proporcione la contraseña del usuario de destino. En el caso del usuario root, tener su contraseña distribuida o conocida por (muchos) usuarios pone en riesgo el sistema y es una muy mala práctica de seguridad. El uso básico de `su` es `su - *nombre-de-usuario-destino*`. Sin embargo, al cambiar a root, el nombre de usuario de destino es opcional:

```
carol@debian:~$ su - root
Password:
root@debian:~# exit
logout
carol@debian:~$ su -
Password:
root@debian:~#
```

El uso del guión (-) garantiza que se cargue el entorno del usuario de destino. Sin el, se mantendrá el entorno del usuario anterior:

```
carol@debian:~$ su
Password:
root@debian:/home/carol#
```

Por otro lado, está el comando `sudo`, con el que se puede ejecutar un comando como usuario root - o cualquier otro usuario. Desde una perspectiva de seguridad, `sudo` es una opción mucho mejor que `su` ya que presenta dos ventajas principales: Para ejecutar un comando como root, no se necesita la contraseña del usuario root, sino sólo la del usuario que lo invoca en cumplimiento de una política de seguridad. La política de seguridad por defecto es `sudoers` como se especifica en `/etc/sudoers` y `/etc/sudoers.d/*`.

1. `sudo` le permite ejecutar comandos individuales con privilegios elevados en lugar de lanzar un nuevo subshell para root como hace `su`.

El uso básico de `sudo` es `sudo -u` *`nombredeusuario comando`*. Sin embargo, para ejecutar un comando como usuario root, la opción `-u` *`nombredeusuario`* no es necesario:

```
carol@debian:~$ sudo -u mimi whoami
mimi
carol@debian:~$ sudo whoami
root
```

> **NOTE**
>
> `sudoers` utilizará una marca de tiempo por usuario (y por terminal) para el almacenamiento en caché de las credenciales, de forma que pueda utilizar `sudo` sin contraseña durante un período por defecto de quince minutos. Este valor por defecto puede ser modificado añadiendo la opción `timestamp_timeout` como un ajuste `Defaults` en `/etc/sudoers` (por ejemplo `Defaults timestamp_timeout=1` establecerá el tiempo de espera de la caché de credenciales en un minuto).

El archivo `/etc/sudoers`

El archivo de configuración principal de `sudo` es `/etc/sudoers` (también existe el directorio `/etc/sudoers.d`). Este es el lugar donde se determinan los privilegios de `sudo` de los usuarios. En otras palabras, aquí se especifica quién puede ejecutar qué comandos como qué usuarios en qué máquinas — así como otras configuraciones. La sintaxis utilizada es la siguiente:

```
carol@debian:~$ sudo less /etc/sudoers
(...)
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
(...)
```

La especificación de privilegios para el usuario root es `ALL=(ALL:ALL) ALL`. Esto se traduce como: el usuario root (`root`) puede iniciar sesión desde todas las máquinas (`ALL`), como todos los usuarios y todos los grupos (`(ALL:ALL)`), y ejecutar todos los comandos (`ALL`). Lo mismo ocurre con los miembros del grupo `sudo` — nótese cómo los nombres de los grupos se identifican con un signo de porcentaje precedente (`%`).

Así, para que el usuario `carol` pueda comprobar el estado de `apache2` desde cualquier host como cualquier usuario o grupo, añadirá la siguiente línea en el fichero `sudoers`:

```
carol   ALL=(ALL:ALL) /usr/bin/systemctl status apache2
```

Puede que quiera ahorrarle a `carol` la molestia de tener que proporcionar su contraseña para ejecutar el comando `systemctl status apache2`. Para ello, puede modificar la línea para que se vea así:

```
carol   ALL=(ALL:ALL) NOPASSWD: /usr/bin/systemctl status apache2
```

Digamos que ahora quiere restringir sus hosts a 192.168.1.7 y permitir que `carol` ejecute `systemctl status apache2` como usuario `mimi`. Usted modificaría la línea de la siguiente manera:

```
carol   192.168.1.7=(mimi) /usr/bin/systemctl status apache2
```

Ahora puede comprobar el estado del servidor web Apache como usuario `mimi`:

```
carol@debian:~$ sudo -u mimi systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
enabled)
   Active: active (running) since Tue 2020-06-09 13:12:19 CEST; 29min ago
(...)
```

Si `carol` fuese promovida a sysadmin y quisiera darle todos los privilegios, el enfoque más fácil sería el de incluirla en el grupo especial `sudo` con `usermod` y la opción `-G` (también es posible querer usar la opción `-a`, que asegura que el usuario no sea removido de cualquier otro grupo al que pudiera pertenecer):

```
root@debian:~# sudo useradd -aG sudo carol
```

NOTE En la familia de distribuciones de Red Hat el grupo `wheel` es la contrapartida del grupo administrativo especial `sudo` de los sistemas Debian.

En lugar de editar `/etc/sudoers` directamente, simplemente debe utilizar el comando `visudo` como root (por ejemplo: `visudo`), que abrirá `/etc/sudoers` utilizando su editor de texto predefinido. Para cambiar el editor de texto por defecto, puedes añadir la opción `editor` como un ajuste `Defaults` en `/etc/sudoers`. Por ejemplo, para cambiar el editor a `nano`, añada la siguiente línea:

```
Defaults    editor=/usr/bin/nano
```

NOTE Como alternativa, puede especificar un editor de texto a través de la variable de entorno `EDITOR` cuando utilice `visudo` (por ejemplo: `EDITOR=/usr/bin/nano visudo`)

Aparte de los usuarios y grupos, también puede hacer uso de los alias en `/etc/sudoers`. Hay tres categorías principales de alias que puedes definir: *alias de host* (`Host_Alias`), *alias de usuario* (`User_Alias`) y *alias de comando* (`Cmnd_Alias`). He aquí un ejemplo:

```
# Especificación de alias de host

Host_Alias SERVERS = 192.168.1.7, server1, server2

# Especificación de alias de usuario

User_Alias REGULAR_USERS = john, mary, alex

User_Alias PRIVILEGED_USERS = mimi, alex

User_Alias ADMINS = carol, %sudo, PRIVILEGED_USERS, !REGULAR_USERS

# Especificación de alias de Cmnd

Cmnd_Alias SERVICES = /usr/bin/systemctl *

# Especificación de los privilegios del usuario
root    ALL=(ALL:ALL) ALL
ADMINS  SERVERS=SERVICES

# Permitir a los miembros del grupo sudo ejecutar cualquier comando
%sudo   ALL=(ALL:ALL) ALL
```

Teniendo en cuenta este archivo de ejemplo `sudoers`, vamos a explicar los tres tipos de alias con un poco más de detalle:

Host aliases

Incluyen una lista separada por comas de nombres de host, direcciones IP, así como redes y netgroups (precedidos por `+`). También se pueden especificar máscaras de red. El alias de host `SERVERS` incluye una dirección IP y dos nombres de host:

```
Host_Alias SERVERS = 192.168.1.7, server1, server2
```

User aliases

Incluyen una lista separada por comas de usuarios especificados como nombres de usuario, grupos (precedidos por `%`) y netgroups (precedidos por `+`). Se pueden excluir usuarios concretos con `!`. El alias de usuario `ADMINS` - por ejemplo - incluye al usuario `carol`, los miembros del grupo `sudo` y aquellos miembros del alias de usuario `PRIVILEGE_USERS` que no pertenecen al alias de usuario `REGULAR_USERS`:

```
User_Alias ADMINS = carol, %sudo, PRIVILEGED_USERS, !REGULAR_USERS
```

Command aliases

Incluyen una lista de comandos y directorios separados por comas. Si se especifica un directorio, se incluirá cualquier archivo de ese directorio, aunque se ignorarán los subdirectorios. El alias de comando `SERVICES` incluye un solo comando con todos sus subcomandos — según lo especificado por el asterisco (*):

```
Cmnd_Alias SERVICES = /usr/bin/systemctl *
```

Como resultado de las especificaciones de alias, la línea `ADMINS SERVERS=SERVICES` bajo la sección `Especificación de privilegios del usuario` se traduce como: todos los usuarios pertenecientes a `ADMINS` pueden usar `sudo` para ejecutar cualquier comando en `SERVICES` en cualquier servidor en `SERVERS`.

> **NOTE** Hay un cuarto tipo de alias que puede incluir en `/etc/sudoers`: Los alias de runa (`Runas_Alias`). Son muy similares a los alias de usuario, pero permiten especificar usuarios por su *identificación de usuario* (UID). Esta característica puede ser conveniente en algunos escenarios.

Ejercicios guiados

1. Complete la siguiente tabla relativa a los permisos especiales:

Permiso especial	**Representación numérica**	**Representación simbólica**	**Buscar archivos con *sólo* ese permiso establecido**
`SUID`			
`SGID`			

2. La visualización de archivos con *sólo* el bit `SUID` o `SGID` activado no suele ser muy práctica. Realice las siguientes tareas para probar que sus búsquedas pueden ser más productivas:
   - Encuentre todos los archivos con el `SUID` (y otros permisos) establecidos en `/usr/bin`:
   - Busque todos los archivos con el `SGID` (y otros permisos) establecidos en `/usr/bin`:
   - Encuentra todos los archivos con el `SUID` o el `SGID` en `/usr/bin`:

3. `chage` permite cambiar la información de caducidad de la contraseña de un usuario. Como root, complete la siguiente tabla proporcionando los comandos correctos en el usuario `mary`:

Significado	**Comandos `chage`**
Hacer que la contraseña sea válida durante 365 días.	
Hacer que el usuario cambie la contraseña en el próximo inicio de sesión.	
Establecer el número mínimo de días entre cambios de contraseña a 1.	
Desactivar la caducidad de la contraseña.	
Permitir al usuario cambiar su contraseña en cualquier momento.	

Significado	**Comandos `chage`**
Establecer el período de advertencia en 7 días y la fecha de caducidad de la cuenta en el 20 de agosto de 2050.	
Imprimir la información de caducidad de la contraseña del usuario.	

4. Complete la siguiente tabla con la utilidad de red apropiada:

Acción	**Comando(s)**
Muestra archivos de red para el host `192.168.1.55` en el puerto `22` usando `lsof`.	
Muestra los procesos que acceden al puerto por defecto del servidor web Apache en su máquina con `fuser`.	
Muestra todos los sockets *udp* que están escuchando en su máquina usando `netstat`.	
Escanea los puertos del `80` al `443` en el host `192.168.1.55` usando `nmap`.	

5. Realice las siguientes tareas relacionadas con el *tamaño del conjunto residente* (*RSS*) y el `ulimit` como un usuario normal:

   - Mostrar límites *flexibles* en el *máximo de RSS*:

   - Mostrar límites *estrictos* en el *máximo RSS*:

   - Establecer los límites *flexibles* en el *máximo RSS* a 5.000 kilobytes:

   - Establecer los límites *estrictos* del *máximo RSS* a 10.000 kilobytes:

   - Por último, intente aumentar el límite *estricto* del *máximo RSS* hasta 15.000 kilobytes. ¿Puede hacerlo? ¿Por qué?

6. Considere la siguiente línea de salida del comando `last` y responda a las preguntas:

```
carol    pts/0        192.168.1.4      Sun May 31 14:16 - 14:22  (00:06)
```

   - ¿Se conectó `carol` desde un host remoto? ¿Por qué?

   - ¿Cuánto duró la sesión de `carol`?

   - ¿Se conectó `carol` a través de un verdadero terminal clásico basado en texto? ¿Por qué?

7. Considere el siguiente extracto de `/etc/sudoers` y responda a la siguiente pregunta.

```
# Especificación de alias de host

Host_Alias SERVERS = 192.168.1.7, server1, server2

# Especificación de alias de usuario

User_Alias REGULAR_USERS = john, mary, alex

User_Alias PRIVILEGED_USERS = mimi, alex

User_Alias ADMINS = carol, %sudo, PRIVILEGED_USERS, !REGULAR_USERS

# Especificación de alias de Cmnd

Cmnd_Alias WEB_SERVER_STATUS = /usr/bin/systemctl status apache2

# User privilege specification
root    ALL=(ALL:ALL) ALL
ADMINS  SERVERS=WEB_SERVER_STATUS

# Permitir a los miembros del grupo sudo ejecutar cualquier comando
%sudo   ALL=(ALL:ALL) ALL
```

¿Puede `alex` comprobar el estado del servidor web Apache en cualquier host? ¿Por qué?

Ejercicios de exploración

1. Además del `SUID` y el `SGID`, existe un tercer permiso especial: el *sticky bit*. Actualmente se utiliza sobre todo en directorios como `/tmp` para evitar que los usuarios habituales borren o muevan archivos que no sean los suyos. Realice las siguientes tareas:
   - Establecer el *sticky bit* en `~/temporal`:
   - Buscar directorios con el *sticky bit* (y cualquier otro permiso) establecido en su directorio principal:
   - Desactivar el *sticky bit* en `~/temporal`:
2. Cuando la contraseña de un usuario está bloqueada mediante `passwd -l` *`nombre`* o `usermod -L` *`nombre`*, ¿cómo se puede saber mirando en `/etc/shadow`?
3. ¿Cuál es la contraparte del comando `usermod` a `chage -E` *`date username`* o `chage --expiredate` *`date username`*?
4. Proporcione dos comandos diferentes `nmap` para escanear todos los 65535 puertos en localhost:

Resumen

En esta lección ha aprendido a realizar una serie de tareas de administración de la seguridad. Se han cubierto los siguientes temas:

- Buscar archivos con los permisos especiales `SUID` y `SGID`.
- Establecer y cambiar las contraseñas de los usuarios y manejar la información de caducidad de las contraseñas.
- Usar una serie de utilidades de red para descubrir puertos abiertos en hosts/redes.
- Establecer límites en los recursos del sistema.
- Verificar los usuarios que han entrado en el sistema o que están actualmente conectados.
- Uso y configuración básica de `sudo` (a través del archivo `/etc/sudoers`).

Comandos y archivos tratados en esta lección:

`find`
: Busca archivos en una jerarquía de directorios.

`passwd`
: Cambia la contraseña del usuario.

`chmod`
: Cambia los bits del modo de archivo.

`chage`
: Modifica la información de caducidad de la contraseña del usuario.

`lsof`
: Lista los archivos abiertos.

`fuser`
: Identifica los procesos que utilizan archivos o sockets.

`netstat`
: Imprime las conexiones de red.

`nmap`
: Herramienta de exploración de redes y escáner de puertos.

`ulimit`
Muestra y establece los límites de los usuarios.

`/etc/security/limits.conf`
Archivo de configuración para aplicar restricciones a los usuarios.

`last`
Imprime un listado de los últimos usuarios conectados.

`lastb`
Imprime un listado de los intentos de inicio de sesión incorrectos.

`/var/log/wtmp`
Base de datos de los inicios de sesión de los usuarios.

`who`
Muestra quién está conectado.

`w`
Muestra quién está conectado y qué está haciendo.

`su`
Cambiar de usuario o convertirse en superusuario.

`sudo`
Ejecuta un comando como otro usuario (incluyendo el superusuario).

`/etc/sudoers`
Archivo de configuración por defecto para la política de seguridad `sudo`.

Respuesta a los ejercicios guiados

1. Complete la siguiente tabla relativa a los permisos especiales:

Permiso especial	**Representación numérica**	**Representación simbólica**	**Buscar archivos con *sólo* ese permiso establecido**
`SUID`	`4000`	`s`,`S`	`find -perm 4000`, `find -perm u+s`
`SGID`	`2000`	`s`,`S`	`find -perm 2000`, `find -perm g+s`

2. La visualización de archivos con *sólo* el bit `SUID` o `SGID` activado no suele ser muy práctica. Realice las siguientes tareas para probar que sus búsquedas pueden ser más productivas:

   - Encuentre todos los archivos con el `SUID` (y otros permisos) establecidos en `/usr/bin`:

     `find /usr/bin -perm -4000` or `find /usr/bin -perm -u+s`

   - Busque todos los archivos con el `SGID` (y otros permisos) establecidos en `/usr/bin`:

     `find /usr/bin -perm -2000` or `find /usr/bin -perm -g+s`

   - Encuentre todos los archivos con el `SUID` o el `SGID` en `/usr/bin`:

     `find /usr/bin -perm /6000`

3. `chage` permite cambiar la información de caducidad de la contraseña de un usuario. Como root, complete la siguiente tabla proporcionando los comandos correctos en el usuario `mary`:

Significado	**Comandos `chage`**
Hacer que la contraseña sea válida durante 365 días.	`chage -M 365 mary`, `chage --maxdays 365 mary`
Hacer que el usuario cambie la contraseña en el próximo inicio de sesión.	`chage -d 0 mary`, `chage --lastday 0 mary`
Establecer el número mínimo de días entre cambios de contraseña a 1.	`chage -m 1 mary`, `chage --mindays 1 mary`
Desactivar la caducidad de la contraseña.	`chage -M 99999 mary`, `chage --maxdays 99999 mary`

Significado	**Comandos** `chage`
Permitir al usuario cambiar su contraseña en cualquier momento.	`chage -m 0 mary`, `chage --mindays 0 mary`
Establecer el período de advertencia en 7 días y la fecha de caducidad de la cuenta en el 20 de agosto de 2050.	`chage -W 7 -E 2050-08-20 mary`, `chage --warndays 7 --expiredate 2050-08-20 mary`
Imprimir la información de caducidad de la contraseña del usuario.	`chage -l mary`, `chage --list mary`

4. Complete la siguiente tabla con la utilidad de red apropiada:

Acción	**Comando(s)**
Muestra archivos de red para el host `192.168.1.55` en el puerto `22` usando `lsof`.	`lsof -i@192.168.1.55:22`
Muestra los procesos que acceden al puerto por defecto del servidor web Apache en su máquina con `fuser`.	`fuser -vn tcp 80`, `fuser --verbose --namespace tcp 80`
Muestra todos los sockets *udp* que están escuchando en su máquina usando `netstat`.	`netstat -lu`,`netstat --listening --udp`
Escanea los puertos del `80` al `443` en el host `192.168.1.55` usando `nmap`.	`nmap -p 80-443 192.168.1.55`

5. Realice las siguientes tareas relacionadas con el *tamaño del conjunto residente* (*RSS*) y el `ulimit` como un usuario normal:

   - Mostrar límites *flexibles* en el *máximo de RSS*:

     `ulimit -m`, `ulimit -Sm`

   - Mostrar límites *estrictos* en el *máximo RSS*:

     `ulimit -Hm`

   - Establecer los límites *flexibles* en el *máximo RSS* a 5.000 kilobytes:

     `ulimit -Sm 5000`

   - Establecer los límites *estrictos* del *máximo RSS* a 10.000 kilobytes:

`ulimit -Hm 10000`

- Por último, intenta aumentar el límite *estricto* del *máximo RSS* hasta 15.000 kilobytes. ¿Puede hacerlo? ¿Por qué?

  No. Una vez establecidos, los usuarios habituales no pueden aumentar los límites estrictos.

6. Considere la siguiente línea de salida del comando `last` y responda a las preguntas:

```
carol    pts/0        192.168.1.4      Sun May 31 14:16 - 14:22  (00:06)
```

- ¿Se conectó `carol` desde un host remoto? ¿Por qué?

  Sí, la dirección IP del host remoto está en la tercera columna.

  *¿Cuánto duró la sesión de `carol`?

  Seis minutos (como se muestra en la última columna).

- ¿Se conectó `carol` a través de un verdadero terminal clásico basado en texto? ¿Por qué?

  No, `pts/0` en la segunda columna indica que la conexión se realizó a través de un emulador de terminal gráfico (también conocido como *Pseudo Terminal Slave*).

7. Considere el siguiente extracto de `/etc/sudoers` y responda a la siguiente pregunta.

```
# Especificación de alias de host

Host_Alias SERVERS = 192.168.1.7, server1, server2

# Especificación de alias de usuario

User_Alias REGULAR_USERS = john, mary, alex

User_Alias PRIVILEGED_USERS = mimi, alex

User_Alias ADMINS = carol, %sudo, PRIVILEGED_USERS, !REGULAR_USERS

# Especificación de alias de Cmnd

Cmnd_Alias WEB_SERVER_STATUS = /usr/bin/systemctl status apache2

# User privilege specification
root    ALL=(ALL:ALL) ALL
ADMINS  SERVERS=WEB_SERVER_STATUS

# Permitir a los miembros del grupo sudo ejecutar cualquier comando
%sudo   ALL=(ALL:ALL) ALL
```

¿Puede `alex` comprobar el estado del servidor web Apache en cualquier host? ¿Por qué?

No, ya que es miembro de `REGULAR_USERS` y ese grupo de usuarios está excluido de `ADMINS`; los únicos usuarios (aparte de `carol`, miembros del grupo `sudo` y `root`) que pueden ejecutar `systemctl status apache2` en los `SERVERS`.

Respuestas a los ejercicios de exploración

1. Además del `SUID` y el `SGID`, existe un tercer permiso especial: el *sticky bit*. Actualmente se utiliza sobre todo en directorios como `/tmp` para evitar que los usuarios habituales borren o muevan archivos que no sean los suyos. Realice las siguientes tareas:

   - Establecer el *sticky bit* en `~/temporal`:

     `chmod +t temporal`, `chmod 1755 temporal`

   - Buscar directorios con el *sticky bit* (y cualquier otro permiso) establecido en su directorio principal:

     `find ~ -perm -1000`, `find ~ -perm /1000`

   - Desactivar el *sticky bit* en `~/temporal`:

     `chmod -t temporal`, `chmod 0755 temporal`

2. Cuando la contraseña de un usuario está bloqueada mediante `passwd -l` *`nombre`* o `usermod -L` *`nombre`*, ¿cómo se puede saber mirando en `/etc/shadow`?

   Aparecerá un signo de exclamación en el segundo campo, justo después del nombre de acceso del usuario afectado (Ejemplo: `mary:!$6$gOg9xJgv…`).

3. ¿Cuál es la contraparte del comando `usermod` a `chage -E` *`date username`* o `chage --expiredate` *`date username`*?

   `usermod -e` *`date username`*, `usermod --expiredate` *`date username`*

4. Proporcione dos comandos diferentes `nmap` para escanear todos los 65535 puertos en localhost:

   `nmap -p 1-65535 localhost` and `nmap -p- localhost`



110.2 Configuración de la seguridad del sistema

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 110.2

Importancia

3

Áreas de conocimiento clave

- Conocer el oscurecimiento de contraseña (shadow passwords) y su funcionamiento.
- Desactivar servicios de red que no estén en uso.
- Entender el papel de TCP wrappers.

Lista parcial de archivos, términos y utilidades

- `/etc/nologin`
- `/etc/passwd`
- `/etc/shadow`
- `/etc/xinetd.d/`
- `/etc/xinetd.conf`
- `systemd.socket`
- `/etc/inittab`
- `/etc/init.d/`
- `/etc/hosts.allow`
- `/etc/hosts.deny`



110.2 Lección 1

Certificación:	LPIC-1
Versión:	5.0
Tema:	110 Seguridad
Objetivo:	110.2 Configuración de la seguridad del sistema
Lección:	1 de 1

Introducción

En este capítulo se explican cuatro formas básicas de mejorar la seguridad del host:

1. Algunos comandos básicos y ajustes de configuración para mejorar la seguridad de la autenticación con contraseñas *shadow*.
2. Cómo utilizar los superdaemons para escuchar las conexiones de red entrantes.
3. Comprobación de los servicios de red en busca de demonios innecesarios.
4. TCP wrappers como una especie de firewall simple.

Mejorar la seguridad de la autenticación con shadow password

Los componentes básicos de los datos de la cuenta de un usuario se almacenan en el archivo `/etc/passwd`. Este archivo contiene siete campos: nombre de inicio de sesión, userid, groupid, contraseña, comentario (también conocido como GECOS), ubicación del directorio principal y por último, el shell por defecto. Una forma sencilla de recordar el orden de estos campos es pensar en el proceso de inicio de sesión de un usuario: primero se introduce un nombre de inicio de sesión, en segundo lugar el sistema lo asignará a un userid (uid) y en tercer lugar a un groupid (gid). El cuarto

paso pide una contraseña, el quinto busca el comentario, el sexto introduce el directorio personal del usuario y el séptimo paso establece el shell por defecto.

Aunque en los sistemas modernos la contraseña ya no se almacena en el archivo `/etc/passwd`. En su lugar, el campo de la contraseña sólo contiene una `x` minúscula. El archivo `/etc/passwd` tiene que ser legible por todos los usuarios. Por lo tanto, no es una buena idea almacenar contraseñas allí. La `x` indica que la contraseña encriptada (hash) se almacena en el archivo `/etc/shadow`. Este archivo no debe ser legible para todos los usuarios.

La configuración de las contraseñas se hace con los comandos `passwd` y `chage`. Ambos comandos cambiarán la entrada para el usuario `emma` en el archivo `/etc/shadow`. Como superusuario puede configurar la contraseña para el usuario `emma` con el siguiente comando:

```
$ sudo passwd emma
New password:
Retype new password:
passwd: password updated successfully
```

A continuación, se le pedirá dos veces que confirme la nueva contraseña.

Para listar el tiempo de expiración de la contraseña y otros ajustes de expiración de la contraseña para el usuario `emma` utilice:

```
$ sudo chage -l emma
Last password change                    : Apr 27, 2020
Password expires                    : never
Password inactive                   : never
Account expires                     : never
Minimum number of days between password change      : 0
Maximum number of days between password change      : 99999
Number of days of warning before password expires   : 7
```

Para evitar que el usuario `emma` se registre en el sistema, el superusuario puede establecer una fecha de caducidad de la contraseña que preceda a la fecha actual. Por ejemplo, si la fecha de hoy fuera 2020-03-27, podría caducar la contraseña del usuario utilizando una fecha más antigua:

```
$ sudo chage -E 2020-03-26 emma
```

Como alternativa, el superusuario puede utilizar:

```
$ sudo passwd -l emma
```

para bloquear la cuenta temporalmente mediante la opción `l` de `passwd`. Para probar los efectos de estos cambios, intente iniciar sesión como la cuenta `emma`:

```
$ sudo login emma
Password:
Your account has expired; please contact your system administrator

Authentication failure
```

Para evitar que todos los usuarios, excepto el usuario root, inicien sesión en el sistema temporalmente, el superusuario puede crear un archivo llamado `/etc/nologin`. Este archivo puede contener un mensaje para los usuarios notificándoles por qué no pueden iniciar sesión (por ejemplo, notificaciones de mantenimiento del sistema). Para más detalles vea `man 5 nologin`. Tenga en cuenta que también hay un comando `nologin` que se puede utilizar para evitar un inicio de sesión cuando se establece como el shell por defecto para un usuario. Por ejemplo:

```
$ sudo usermod -s /sbin/nologin emma
```

Para más detalles consulte las páginas de manual del comando.

Cómo utilizar un superdemonio para escuchar las conexiones de red entrantes

Los servicios de red, como los servidores web, los de correo electrónico y de impresión, suelen ejecutarse como independiente que escucha en un puerto de red dedicado. Todos estos servicios se ejecutan uno al lado del otro. En un sistema clásico basado en Sys-V-init cada uno de estos servicios puede ser controlado por el comando `service`. En los sistemas actuales basados en systemd se utiliza `systemctl` para gestionar el servicio. En épocas anteriores, la disponibilidad de recursos informáticos era mucho menor. Ejecutar muchos servicios en modo autónomo no era una buena opción. En su lugar, se utilizaba el llamado superdemonio, que escuchaba las conexiones de red entrantes e iniciaba el servicio apropiado a petición. Este método de crear una conexión de red requería un poco más de tiempo. Los superdemonios más conocidos son `inetd` y `xinetd`. En los sistemas actuales basados en systemd la unidad `systemd.socket` se puede utilizar de forma similar. En esta sección utilizaremos `xinetd` para interceptar las conexiones al demonio `sshd` y arrancar este demonio a petición para demostrar cómo se utiliza el superdemonio.

Antes de configurar el servicio `xinetd` es necesario realizar algunos preparativos. No importa si utiliza un sistema basado en Debian o en Red Hat. Aunque estas explicaciones han sido probadas con

Debian/GNU Linux 9.9 deberían funcionar en cualquier sistema Linux actual que cuente con systemd, sin ningún cambio significativo. Primero asegúrese de que los paquetes `openssh-server` y `xinetd` están instalados. Ahora verifique que el servicio SSH funciona con:

```
$ systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
   Active: active (running) since Mon 2020-04-27 09:33:48 EDT; 3h 11min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 430 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 460 (sshd)
    Tasks: 1 (limit: 1119)
   Memory: 5.3M
   CGroup: /system.slice/ssh.service
           └─460 /usr/sbin/sshd -D
```

Compruebe también que el servicio SSH está escuchando en su puerto de red estándar 22:

```
# lsof -i :22
COMMAND  PID USER   FD   TYPE   DEVICE SIZE/OFF NODE NAME
sshd    1194 root    3u  IPv4 16053268      0t0  TCP *:ssh (LISTEN)
sshd    1194 root    4u  IPv6 16053270      0t0  TCP *:ssh (LISTEN)
```

Finalmente detenga el servicio SSH con:

```
$ sudo systemctl stop sshd.service
```

En el caso de que quiera hacer este cambio permanente utilice `systemctl disable sshd.service`.

Ahora puede crear el archivo de configuración de xinetd `/etc/xinetd.d/ssh` con algunos ajustes básicos:

```
service ssh
{
    disable     = no
    socket_type = stream
    protocol    = tcp
    wait        = no
    user        = root
    server      = /usr/sbin/sshd
    server_args     = -i
    flags       = IPv4
    interface   = 192.168.178.1
}
```

Reinicie el servicio xinetd con:

```
$ sudo systemctl restart xinetd.service
```

Compruebe qué servicio está escuchando ahora las conexiones SSH entrantes.

```
$ sudo lsof -i :22
COMMAND   PID USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
xinetd  24098 root    5u  IPv4 7345141      0t0  TCP 192.168.178.1:ssh (LISTEN)
```

Podemos ver que el servicio xinetd ha tomado el control para el acceso del puerto 22.

Aquí hay algunos detalles más sobre la configuración de `xinetd`. El archivo de configuración principal es `/etc/xinetd.conf`:

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{

# Please note that you need a log_type line to be able to use log_on_success
# and log_on_failure. The default is the following :
# log_type = SYSLOG daemon info

}

includedir /etc/xinetd.d
```

Además de la configuración por defecto, sólo hay una directiva para establecer un directorio de inclusión. En este directorio puede establecer un único fichero de configuración para cada servicio que quiera que sea gestionado por `xinetd`. Nosotros hemos hecho esto para el servicio SSH y hemos llamado al fichero `/etc/xinetd.d/ssh`. Los nombres de los ficheros de configuración pueden ser elegidos arbitrariamente, excepto los nombres de ficheros que contengan un punto (.) o que terminen con una tilde (~). Pero es una práctica generalizada nombrar el fichero con el nombre del servicio que se quiere configurar.

Algunos archivos de configuración en el directorio `/etc/xinet.d/` ya son proporcionados por la distribución:

```
$ ls -l /etc/xinetd.d
total 52
-rw-r--r-- 1 root root 640 Feb  5  2018 chargen
-rw-r--r-- 1 root root 313 Feb  5  2018 chargen-udp
-rw-r--r-- 1 root root 502 Apr 11 10:18 daytime
-rw-r--r-- 1 root root 313 Feb  5  2018 daytime-udp
-rw-r--r-- 1 root root 391 Feb  5  2018 discard
-rw-r--r-- 1 root root 312 Feb  5  2018 discard-udp
-rw-r--r-- 1 root root 422 Feb  5  2018 echo
-rw-r--r-- 1 root root 304 Feb  5  2018 echo-udp
-rw-r--r-- 1 root root 312 Feb  5  2018 servers
-rw-r--r-- 1 root root 314 Feb  5  2018 services
-rw-r--r-- 1 root root 569 Feb  5  2018 time
-rw-r--r-- 1 root root 313 Feb  5  2018 time-udp
```

Estos archivos pueden ser utilizados como plantillas en el raro caso de que tenga que utilizar algunos

servicios heredados como `daytime`, una implementación muy temprana de un servidor de tiempo. Todos estos archivos de plantilla contienen la directiva `disable = yes`.

Aquí hay más detalles sobre las directivas usadas en el archivo de ejemplo `/etc/xinetd.d/ssh` para ssh arriba.

```
service ssh
{
    disable     = no
    socket_type = stream
    protocol    = tcp
    wait        = no
    user        = root
    server      = /usr/sbin/sshd
    server_args     = -i
    flags       = IPv4
    interface   = 192.168.178.1
}
```

`service`

Muestra el servicio que xinetd debe controlar. Puede utilizar un número de puerto, como el 22, o el nombre asignado al número de puerto en `/etc/services`, por ejemplo `ssh`.

`{`

Los ajustes detallados comienzan con una llave de apertura.

`disable`

Para activar esta configuración, póngala en `no`. Si quiere desactivar la configuración temporalmente, puede ponerla en `yes`.

`socket_type`

Puede elegir `stream` para sockets TCP o `dgram` para sockets UDP y más.

`protocol`

Elija entre TCP o UDP.

`wait`

En el caso de las conexiones TCP, este valor suele ser `no`.

`user`

El servicio iniciado en esta línea será propiedad de este usuario.

`server`

Ruta completa del servicio que debe ser iniciado por `xinetd`.

`server_args`

Aquí puede añadir opciones para el servicio. Si es iniciado por un super-servidor, muchos servicios requieren una opción especial. Para SSH esta sería la opción `-i`.

`flags`

Puede elegir IPv4, IPv6 y otros.

`interface`

La interfaz de red que `xinetd` debe controlar. Nota: también puede elegir la directiva `bind`, que no es más que un sinónimo de `interfaz`.

`}`

Termina con un corchete de cierre.

Los sucesores de los servicios iniciados por el super-servidor `xinetd` son unidades de socket systemd. Configurar una unidad de socket systemd es muy sencillo y fácil, porque ya existe una unidad de socket systemd predefinida para SSH. Asegúrese de que los servicios `xinetd` y SSH no se están ejecutando.

Ahora sólo tiene que iniciar la unidad de socket SSH:

```
$ sudo systemctl start ssh.socket
```

Para comprobar qué servicio está ahora escuchando en el puerto 22 utilizamos de nuevo `lsof`. Observe que aquí se ha utilizado la opción `-P` para mostrar el número de puerto en lugar del nombre del servicio en la salida:

```
$ sudo lsof -i :22 -P
COMMAND PID USER   FD   TYPE   DEVICE SIZE/OFF NODE NAME
systemd   1 root   57u  IPv6 14730112      0t0  TCP *:22 (LISTEN)
```

Para que esta sesión sea completa, debe intentar iniciar sesión en su servidor con un cliente SSH de su elección.

TIP En caso de que `systemctl start ssh.socket` no funcione con su distribución, pruebe con `systemctl start sshd.socket`.

Comprobación de servicios en busca de daemons innecesarios

Por razones de seguridad, así como para controlar los recursos del sistema, es importante tener una visión general de los servicios que se están ejecutando. Los servicios innecesarios y no utilizados deben ser desactivados. Por ejemplo, si no necesita distribuir páginas web, no es necesario ejecutar un servidor web como Apache o nginx.

En los sistemas basados en SyS-V-init puede comprobar el estado de todos los servicios con lo siguiente:

```
$ sudo service --status-all
```

Verifique si cada uno de los servicios listados en la salida del comando son necesarios y desactive todos los servicios innecesarios con (para sistemas basados en Debian):

```
$ sudo update-rc.d SERVICE-NAME remove
```

O en los sistemas basados en Red Hat se utilizaría:

```
$ sudo chkconfig SERVICE-NAME off
```

En los sistemas modernos basados en systemd podemos utilizar lo siguiente para listar todos los servicios en ejecución:

```
$ systemctl list-units --state active --type service
```

A continuación, desactivaría cada unidad de servicio innecesaria con:

```
$ sudo systemctl disable UNIT --now
```

Este comando detendrá el servicio y lo eliminará de la lista de servicios, para evitar que se inicie en el próximo arranque del sistema.

Además, para obtener un estudio de los servicios de red en escucha, puede utilizar `netstat` en sistemas antiguos (siempre que tenga instalado el paquete `net-tools`):

```
$ netstat -ltu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp6       0      0 [::]:http               [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
```

O en los sistemas modernos, puede utilizar el comando equivalente `ss` (para “socket services”):

```
$ ss -ltu
Netid      State          Recv-Q         Send-Q      Local Address:Port
Peer Address:Port
udp        UNCONN         0              0                 0.0.0.0:bootpc
0.0.0.0:*
tcp        LISTEN         0              128               0.0.0.0:ssh
0.0.0.0:*
tcp        LISTEN         0              80              127.0.0.1:mysql
0.0.0.0:*
tcp        LISTEN         0              128                     *:http
*:*
tcp        LISTEN         0              128                  [::]:ssh
[::]:*
```

TCP Wrappers como una especie de Firewall simple

En los tiempos en que no había cortafuegos disponibles para Linux, se utilizaban TCP Wrappers para asegurar las conexiones de red en un host. Hoy en día muchos programas ya no obedecen a los TCP wrappers. En las distribuciones recientes basadas en Red Hat (por ejemplo, Fedora 29) el soporte de TCP wrappers ha sido eliminado completamente. Para dar soporte a los sistemas Linux heredados que todavía utilizan TCP wrappers, es útil tener algunos conocimientos básicos sobre esta tecnología en particular.

Una vez más utilizaremos el servicio SSH como ejemplo básico. El servicio en nuestro host de ejemplo debe ser accesible desde la red local solamente. Primero, comprobamos si el demonio SSH utiliza la biblioteca `libwrap` que ofrece soporte a TCP wrappers:

```
$ ldd /usr/sbin/sshd | grep "libwrap"
        libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007f91dbec0000)
```

Ahora, añadimos la siguiente línea en el archivo `/etc/hosts.deny`:

```
sshd: ALL
```

Finalmente, configuramos una excepción en el archivo `/etc/hosts.allow` para las conexiones SSH desde la red local:

```
sshd: LOCAL
```

Los cambios tienen efecto inmediato, no es necesario reiniciar ningún servicio. Puede comprobarlo con el cliente `ssh`.

Ejercicios guiados

1. ¿Cómo se puede desbloquear la cuenta `emma` previamente bloqueada?

2. Anteriormente la cuenta `emma` tenía una fecha de caducidad establecida. ¿Cómo se puede fijar la fecha de caducidad en `nunca`?

3. Imagine que el servicio de impresión CUPS que gestiona los trabajos de impresión no es necesario en su servidor. ¿Cómo puede desactivar el servicio de forma permanente? ¿Cómo puede comprobar que el puerto correspondiente ya no está activo?

4. Ha instalado el servidor web nginx. Cómo puede comprobar si nginx admite TCP wrappers ?

Ejercicios de exploración

1. Compruebe si la existencia del archivo `/etc/nologin` impide el inicio de sesión del usuario `root`.

2. ¿La existencia del archivo `/etc/nologin` impide el inicio de sesión sin contraseña con claves SSH?

3. ¿Qué sucede en el inicio de sesión, cuando el archivo `/etc/nologin` contiene esta línea de texto `login currently is not possible` solamente?

4. ¿Puede un usuario ordinario `emma` obtener información sobre el usuario `root` contenida en el fichero `/etc/passwd` por ejemplo con el comando `grep root /etc/passwd`?

5. ¿Puede un usuario ordinario `emma` obtener información sobre el usuario `root` contenida en el fichero `/etc/passwd` por ejemplo con el comando `grep root /etc/passwd`?

6. ¿Qué pasos hay que dar para habilitar y comprobar que el servicio heredado `daytime` sea manejado por xinetd? Tenga en cuenta que esto es sólo un ejercicio de exploración no hacer esto en un entorno de producción.

Resumen

En esta lección aprendió:

1. En qué archivo se almacenan las contraseñas, así como algunos ajustes de seguridad de las contraseñas, por ejemplo, el tiempo de caducidad.
2. El propósito del superdemonio `xinetd` y cómo hacerlo funcionar e iniciar el servicio `sshd` bajo demanda.
3. Cómo comprobar qué servicios de red se están ejecutando y cómo desactivar los servicios innecesarios.
4. La utilización de TCP Wrappers como una especie de cortafuegos simple.

Los comandos utilizados en esta lección incluyen:

`chage`
: Cambiar la edad de la contraseña de un usuario.

`chkconfig`
: Un comando clásico utilizado inicialmente en los sistemas basados en Red Hat para establecer si un servicio se iniciará en el momento del arranque o no.

`netstat`
: Una utilidad clásica (ahora en el paquete `net-tools`) que mostrará los demonios que acceden a los puertos de red en un sistema y su uso.

`nologin`
: Un comando que se puede utilizar en lugar del shell de un usuario para evitar que inicie sesión.

`passwd`
: Se utiliza para crear o cambiar la contraseña de un usuario.

`service`
: Método más antiguo para controlar el estado de un demonio, como detener o iniciar un servicio.

`ss`
: El equivalente moderno a `netstat`, pero también muestra más información sobre los distintos sockets en uso en el sistema.

`systemctl`

El comando del sistema utilizado para controlar varios aspectos de los servicios y sockets en un ordenador utilizando systemd.

`update-rc.d`

Un comando clásico similar a `chkconfig` que habilita o deshabilita el arranque de un sistema en las distribuciones basadas en Debian.

`xinetd`

Un superdemonio que puede controlar el acceso a un servicio de red bajo demanda, dejando así un servicio inactivo hasta que se le pida que realice alguna tarea.

Respuestas a los ejercicios guiados

1. ¿Cómo se puede desbloquear la cuenta `emma` previamente bloqueada?

   El superusuario puede ejecutar `passwd -u emma` para desbloquear la cuenta.

2. Anteriormente la cuenta `emma` tenía una fecha de caducidad establecida. ¿Cómo se puede fijar la fecha de caducidad en `nunca`?

   El superusuario puede utilizar `chage -E -1 emma` para fijar la fecha de caducidad en nunca. Esta configuración se puede comprobar con `chage -l emma`.

3. Imagine que el servicio de impresión CUPS que gestiona los trabajos de impresión no es necesario en su servidor. ¿Cómo puede desactivar el servicio de forma permanente? ¿Cómo puede comprobar que el puerto correspondiente ya no está activo?

   Como superusuario

   ```
   systemctl disable cups.service --now
   ```

   Ahora puede comprobar

   ```
   netstat -l | grep ":ipp "` o `ss -l | grep ":ipp "
   ```

4. Ha instalado el servidor web nginx. Cómo puede comprobar si nginx admite TCP wrappers ?

   El comando

   ```
   ldd /usr/sbin/nginx | grep "libwrap"
   ```

   mostrará una entrada en caso de que nginx soporte TCP wrappers.

Respuestas a los ejercicios de exploración

1. Compruebe si la existencia del archivo `/etc/nologin` impide el inicio de sesión del usuario `root`?

   El usuario `root` aún puede iniciar sesión.

2. ¿La existencia del archivo `/etc/nologin` impide el inicio de sesión sin contraseña con claves SSH?

   Sí, también se impiden los inicios de sesión sin contraseña.

3. ¿Qué sucede en el inicio de sesión, cuando el archivo `/etc/nologin` contiene esta línea de texto `login currently is not possible` solamente?

   Se mostrará el mensaje `login currently is not possible` y se impedirá el inicio de sesión.

4. ¿Puede un usuario ordinario `emma` obtener información sobre el usuario `root` contenida en el fichero `/etc/passwd` por ejemplo con el comando `grep root /etc/passwd`?

   Sí, porque todos los usuarios tienen permiso de lectura para este archivo.

5. ¿Puede un usuario ordinario `emma` obtener información sobre el usuario `root` contenida en el fichero `/etc/passwd` por ejemplo con el comando `grep root /etc/passwd`?

   No, porque los usuarios normales no tienen permiso de lectura para este archivo.

6. ¿Qué pasos hay que dar para habilitar y comprobar que el servicio heredado `daytime` sea manejado por xinetd? Tenga en cuenta que esto es sólo un ejercicio de exploración no hacer esto en un entorno de producción.

   Primero, cambie el archivo `/etc/xinetd.d/daytime` y establezca la directiva `disable = no`. Segundo, reinicie el servicio xinetd `systemctl restart xinetd.service` (o `service xinetd restart` en sistemas con SyS-V-Init). Ahora puede comprobar si funciona `nc localhost daytime`. En lugar de `nc` también puede utilizar `netcat`.



110.3 Protección de datos mediante cifrado

Referencia al objetivo del LPI

LPIC-1 version 5.0, Exam 102, Objective 110.3

Importancia

4

Áreas de conocimiento clave

- Configuración y uso básicos del cliente OpenSSH 2.
- Entender el papel que desempeñan las claves del servidor OpenSSH 2.
- Configuración y uso básicos de GnuPG, incluyendo la revocación de claves.
- Uso de GPG para cifrar, descifrar, firmar y verificar archivos.
- Entender la redirección de puertos a través de túneles SSH (incluyendo los túneles X11).

Lista parcial de archivos, términos y utilidades

- `ssh`
- `ssh-keygen`
- `ssh-agent`
- `ssh-add`
- `~/.ssh/id_rsa` y `id_rsa.pub`
- `~/.ssh/id_dsa` y `id_dsa.pub`
- `~/.ssh/id_ecdsa` y `id_ecdsa.pub`
- `~/.ssh/id_ed25519` y `id_ed25519.pub`

- `/etc/ssh/ssh_host_rsa_key` y `ssh_host_rsa_key.pub`
- `/etc/ssh/ssh_host_dsa_key` y `ssh_host_dsa_key.pub`
- `/etc/ssh/ssh_host_ecdsa_key` y `ssh_host_ecdsa_key.pub`
- `/etc/ssh/ssh_host_ed25519_key` y `ssh_host_ed25519_key.pub`
- `~/.ssh/authorized_keys`
- `ssh_known_hosts`
- `gpg`
- `gpg-agent`
- `~/.gnupg/`



110.3 Lección 1

Versión:	5.0
Tema:	110 Seguridad
Objetivo:	110.3 Protección de datos mediante cifrado
Lección:	1 de 2

Introducción

Asegurar los datos con encriptación es de suma importancia en muchos aspectos de la administración de sistemas de hoy en día, y más aún cuando se trata de acceder a sistemas de forma remota. A diferencia de soluciones inseguras como *telnet*, *rlogin* o *FTP*, el protocolo *SSH* (*Secure Shell*) fue diseñado pensando en la seguridad. Utilizando criptografía de clave pública, autentifica tanto a los hosts como a los usuarios y encripta todo el intercambio de información posterior. Además, SSH puede utilizarse para establecer *túneles de puertos*, lo que -entre otras cosas- permite que un protocolo no cifrado transmita datos a través de una conexión SSH cifrada. La versión actual y recomendada del protocolo SSH es la 2.0. *OpenSSH* es una implementación libre y de código abierto del protocolo SSH.

Esta lección cubrirá la configuración básica del cliente *OpenSSH* así como el papel de las claves del servidor *OpenSSH*. También se discutirá el concepto de túneles de puertos SSH. Utilizaremos dos máquinas con la siguiente configuración:

Rol del equipo	Sistema Operativo	Dirección IP	Hostname	Usuario
Cliente	Debian GNU/Linux 10 (buster)	`192.168.1.55`	`debian`	`carol`
Servidor	openSUSE Leap 15.1	`192.168.1.77`	`halof`	`ina`

Configuración y uso básico del cliente OpenSSH

Aunque el servidor y el cliente de OpenSSH vienen en paquetes separados, normalmente se puede instalar un metapaquete que proporcione ambos a la vez. Para establecer una sesión remota con el servidor SSH se utiliza el comando `ssh`, especificando el usuario con el que se quiere conectar en la máquina remota y la dirección IP o el nombre de la máquina remota. La primera vez que se conecte a una máquina remota recibirá un mensaje como este:

```
carol@debian:~$ ssh ina@192.168.1.77
The authenticity of host '192.168.1.77 (192.168.1.77)' can't be established.
ECDSA key fingerprint is SHA256:5JF7anupYipByCQm2BPvDHRVFJJixeslmppi2NwATYI.
Are you sure you want to continue connecting (yes/no)?
```

Después de teclear `sí` y pulsar Enter, se le pedirá la contraseña del usuario remoto. Si se introduce correctamente, se mostrará un mensaje de advertencia y se iniciará la sesión en el host remoto:

```
Warning: Permanently added '192.168.1.77' (ECDSA) to the list of known hosts.
Password:
Last login: Sat Jun 20 10:52:45 2020 from 192.168.1.4
Have a lot of fun...
ina@halof:~>
```

Los mensajes se explican por sí mismos: como era la primera vez que establecía una conexión con el servidor remoto `192.168.1.77`, su autenticidad no podía ser comprobada con ninguna base de datos. Por lo tanto, el servidor remoto proporcionó una `huella de clave ECDSA` de su clave pública (utilizando la función hash `SHA256`). Una vez aceptada la conexión, la clave pública del servidor remoto se añadía a la base de datos de *hosts conocidos*, permitiendo así la autenticación del servidor para futuras conexiones. Esta lista de claves públicas de *hosts* conocidos se mantiene en el archivo `known_hosts` ubicado en `~/.ssh`:

```
ina@halof:~> exit
logout
Connection to 192.168.1.77 closed.
carol@debian:~$ ls .ssh/
known_hosts
```

Tanto `.ssh` como `known_hosts` fueron creados después de establecer la primera conexión remota. `~/.ssh` es el directorio por defecto para la configuración específica del usuario y la información de autenticación.

NOTE También puede utilizar `ssh` para ejecutar un solo comando en el host remoto y luego volver a su terminal local (por ejemplo: ejecutar `ssh ina@halof ls`).

Si está utilizando el mismo usuario tanto en el host local como en el remoto, no es necesario especificar el nombre de usuario al establecer la conexión SSH. Por ejemplo, si está conectado como usuario `carol` en `debian` y quiere conectarse a `halof` también como usuario `carol`, simplemente escribiría `ssh 192.168.1.77` o `ssh halof` (si el nombre puede ser resuelto):

```
carol@debian:~$ ssh halof
Password:
Last login: Wed Jul  1 23:45:02 2020 from 192.168.1.55
Have a lot of fun...
carol@halof:~>
```

Ahora suponga que establece una nueva conexión remota con un host que casualmente tiene la misma dirección IP que `halof` (algo común si utiliza DHCP en su LAN). Se le advertirá de la posibilidad de un ataque *man-in-the-middle*:

```
carol@debian:~$ ssh john@192.168.1.77
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:KH4q3vP6C7e0SEjyG8Wlz9fVlf+jmWJ5139RBxBh3TY.
Please contact your system administrator.
Add correct host key in /home/carol/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/carol/.ssh/known_hosts:1
  remove with:
  ssh-keygen -f "/home/carol/.ssh/known_hosts" -R "192.168.1.77"
ECDSA host key for 192.168.1.77 has changed and you have requested strict checking.
Host key verification failed.
```

Como no se trata de un ataque *man-in-the-middle*, puede añadir con seguridad la huella de la clave pública del nuevo host a `.ssh/known_hosts`. Como indica el mensaje, primero, puede utilizar el comando `ssh-keygen -f "/home/carol/.ssh/known_hosts" -R "192.168.1.77"` para eliminar la clave *ofensiva* (alternativamente, puede optar por `ssh-keygen -R 192.168.1.77` para eliminar todas las claves pertenecientes a `192.168.1.77` de `~/.ssh/known_hosts`). Entonces, podrá establecer una conexión con el nuevo host.

Inicio de sesión basado en claves

Puede configurar su cliente SSH para que no proporcione ninguna contraseña al iniciar la sesión, sino que utilice claves públicas. Este es el método preferido para conectarse a un servidor remoto vía SSH, ya que es mucho más seguro. Lo primero que tiene que hacer es crear un par de claves en la máquina cliente. Para hacer esto, usará `ssh-keygen` con la opción `-t` especificando el tipo de encriptación deseado (*Elliptic Curve Digital Signature Algorithm* en nuestro caso). A continuación, se le pedirá la ruta para guardar el par de claves (`~/.ssh/` es conveniente, así como la ubicación por defecto) y una frase de contraseña. Aunque la frase de contraseña es opcional, se recomienda encarecidamente utilizarla siempre.

```
carol@debian:~/.ssh$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/carol/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/carol/.ssh/id_ecdsa.
Your public key has been saved in /home/carol/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:tlamD0SaTquPZYdNepwj8XN4xvqmHCbe8g5FKKUfMo8 carol@debian
The key's randomart image is:
+---[ECDSA 256]---+
|      .          |
|     o .         |
|    = o o        |
|     B *         |
|    E B S o      |
|     o & O       |
|      @ ^ =      |
|     *.@ @.      |
|    o.o+B+o      |
+----[SHA256]-----+
```

NOTE Al crear el par de claves, puede pasar a `ssh-keygen` la opción `-b` para especificar el tamaño de la clave en bits (por ejemplo: `ssh-keygen -t ecdsa -b 521`).

El comando anterior produjo dos archivos más en su directorio `~/.ssh`:

```
carol@debian:~/.ssh$ ls
id_ecdsa  id_ecdsa.pub  known_hosts
```

`id_ecdsa`

Esta es su clave privada.

`id_ecdsa.pub`

Esta es su clave pública.

NOTE En la criptografía asimétrica (también conocida como criptografía de clave pública), las claves públicas y privadas están relacionadas matemáticamente entre sí de manera que lo que se cifra con una sólo puede descifrarse con la otra.

Lo siguiente que debe hacer es añadir su clave pública al archivo `~/.ssh/authorized_keys` del usuario con el que quiere iniciar sesión en el host remoto (si el directorio `~/.ssh` no existe todavía, tendrá que crearlo primero). Puede copiar su clave pública en el servidor remoto de varias maneras: usando una memoria USB, a través del comando `scp` - que transferirá el archivo usando SSH - o pasando con un *cat* el contenido de su clave pública y pasándolo a `ssh` de esta manera:

```
carol@debian:~/.ssh$ cat id_ecdsa.pub |ssh ina@192.168.1.77 'cat >>
.ssh/authorized_keys'
Password:
```

Una vez que su clave pública haya sido añadida al archivo `authorized_keys` en el host remoto, puede enfrentarse a dos escenarios cuando intente establecer una nueva conexión:

- Si no ha proporcionado una frase de contraseña al crear el par de claves, se iniciará la sesión automáticamente. Aunque es conveniente, este método puede ser inseguro dependiendo de la situación:

  ```
  carol@debian:~$ ssh ina@192.168.1.77
  Last login: Thu Jun 25 20:31:03 2020 from 192.168.1.55
  Have a lot of fun...
  ina@halof:~>
  ```

- Si proporcionó una frase de contraseña al crear el par de claves, tendrá que introducirla en cada conexión de la misma manera que si fuera una contraseña. Aparte de la clave pública, este método añade una capa extra de seguridad en forma de frase de contraseña y puede — por tanto — considerarse más seguro. Sin embargo, en lo que respecta a la comodidad, es exactamente lo mismo que tener que introducir una contraseña cada vez que se establece una conexión. Si no utiliza una frase de contraseña y alguien consigue obtener su archivo de clave SSH privada, tendría acceso a todos los servidores en los que esté instalada su clave pública.

  ```
  carol@debian:~/.ssh$ ssh ina@192.168.1.77
  Enter passphrase for key '/home/carol/.ssh/id_ecdsa':
  Last login: Thu Jun 25 20:39:30 2020 from 192.168.1.55
  Have a lot of fun...
  ina@halof:~>
  ```

Sin embargo, hay una forma que combina seguridad y comodidad: utilizar el agente de autenticación *SSH* (`ssh-agent`). El agente de autenticación necesita generar su propio shell y mantendrá sus claves privadas -para la autenticación con clave pública- en memoria durante el resto de la sesión. Veamos

cómo funciona con un poco más de detalle:

1. Utilice `ssh-agent` para iniciar un nuevo shell Bash:

```
carol@debian:~/.ssh$ ssh-agent /bin/bash
carol@debian:~/.ssh$
```

2. Utilice el comando `ssh-add` para añadir su clave privada a una zona segura de la memoria. Si proporciona una frase de contraseña al generar el par de claves (lo que se recomienda para mayor seguridad) se le pedirá:

```
carol@debian:~/.ssh$ ssh-add
Enter passphrase for /home/carol/.ssh/id_ecdsa:
Identity added: /home/carol/.ssh/id_ecdsa (carol@debian)
```

Una vez añadida su identidad, podrá iniciar sesión en cualquier servidor remoto en el que esté presente su clave pública sin tener que volver a escribir su frase de acceso. Es una práctica habitual en los ordenadores de sobremesa modernos realizar este comando al arrancar el ordenador, ya que permanecerá en la memoria hasta que se apague el ordenador (o se descargue la clave manualmente).

Completemos esta sección enumerando los cuatro tipos de algoritmos de clave pública que se pueden especificar con `ssh-keygen`:

RSA

Llamado así por sus creadores Ron Rivest, Adi Shamir y Leonard Adleman, fue publicado en 1977. Se considera seguro y sigue siendo muy utilizado en la actualidad. Su tamaño mínimo de clave es de 1024 bits (por defecto es de 2048).

DSA

El Algoritmo de Firma Digital (DSA) ha demostrado ser inseguro y ha sido obviado a partir de OpenSSH 7.0. Las claves DSA deben tener exactamente 1024 bits de longitud.

`ecdsa`

El Algoritmo de Firma Digital de Curva Elíptica es una mejora del DSA y, por tanto, se considera más seguro. Utiliza la criptografía de curva elíptica. La longitud de la clave ECDSA está determinada por uno de los tres tamaños posibles de la curva elíptica en bits: 256, 384 o 521.

`ed25519`

Se trata de una implementación de `EdDSA` — *Algoritmo de Firma Digital de la Curva de Edwardsm* — que utiliza la curva 25519 más fuerte. Se considera la más segura de todas. Todas las claves Ed25519 tienen una longitud fija de 256 bits.

NOTE Si se invoca sin especificar `-t`, `ssh-keygen` generará un par de claves `RSA` por defecto.

El papel de las claves del servidor OpenSSH

El directorio de configuración global de OpenSSH se ubica en el directorio `/etc`:

```
halof:~ # tree /etc/ssh
/etc/ssh
▯▯▯ moduli
▯▯▯ ssh_config
▯▯▯ ssh_host_dsa_key
▯▯▯ ssh_host_dsa_key.pub
▯▯▯ ssh_host_ecdsa_key
▯▯▯ ssh_host_ecdsa_key.pub
▯▯▯ ssh_host_ed25519_key
▯▯▯ ssh_host_ed25519_key.pub
▯▯▯ ssh_host_rsa_key
▯▯▯ ssh_host_rsa_key.pub
▯▯▯ sshd_config

0 directories, 11 files
```

Aparte de `moduli` y los archivos de configuración para el cliente (`ssh_config`) y el servidor (`sshd_config`), encontrará cuatro pares de claves — un par de claves para cada algoritmo soportado — que se crean cuando se instala el servidor *OpenSSH*. Como ya se ha dicho, el servidor utiliza estas *claves de host* para identificarse ante los clientes según sea necesario. Su patrón de nombres es el siguiente:

Claves privadas

`ssh_host_` prefix + *algorithm* + `key` suffix (p. ej.: `ssh_host_rsa_key`)

Claves públicas (o huellas de clave pública)

`ssh_host_` prefix + *algorithm* + `key.pub` suffix (p. ej.: `ssh_host_rsa_key.pub`)

> **NOTE** Una huella digital se crea aplicando una función hash criptográfica a una clave pública. Como son más cortas que las claves a las que se refieren, resultan útiles para simplificar ciertas tareas de gestión de claves.

Los permisos de los archivos que contienen las claves privadas son `0600` o `-rw-------`: sólo pueden ser leídos y escritos por el propietario (`root`). Por otro lado, todos los archivos de claves públicas también son legibles por los miembros del grupo propietario y por todos los demás (`0644` o `-rw-r--r--`):

```
halof:~ # ls -l /etc/ssh/ssh_host_*
-rw------- 1 root root 1381 Dec 21 20:35 /etc/ssh/ssh_host_dsa_key
-rw-r--r-- 1 root root  605 Dec 21 20:35 /etc/ssh/ssh_host_dsa_key.pub
-rw------- 1 root root  505 Dec 21 20:35 /etc/ssh/ssh_host_ecdsa_key
-rw-r--r-- 1 root root  177 Dec 21 20:35 /etc/ssh/ssh_host_ecdsa_key.pub
-rw------- 1 root root  411 Dec 21 20:35 /etc/ssh/ssh_host_ed25519_key
-rw-r--r-- 1 root root   97 Dec 21 20:35 /etc/ssh/ssh_host_ed25519_key.pub
-rw------- 1 root root 1823 Dec 21 20:35 /etc/ssh/ssh_host_rsa_key
-rw-r--r-- 1 root root  397 Dec 21 20:35 /etc/ssh/ssh_host_rsa_key.pub
```

Puede ver las huellas digitales de las claves pasando a `ssh-keygen` la opción `-l`. También debe proporcionar la opción `-f` para especificar la ruta del archivo de claves:

```
halof:~ # ssh-keygen -l -f /etc/ssh/ssh_host_ed25519_key
256 SHA256:8cnPrinC49ZHc+/9Ai5pV+1JfZ4WBRZhd3rDOsc2zlA root@halof (ED25519)
halof:~ # ssh-keygen -l -f /etc/ssh/ssh_host_ed25519_key.pub
256 SHA256:8cnPrinC49ZHc+/9Ai5pV+1JfZ4WBRZhd3rDOsc2zlA root@halof (ED25519)
```

Para ver la huella digital de la llave, así como su arte aleatorio, basta con añadir la opción `-v` de esta manera:

```
halof:~ # ssh-keygen -lv -f /etc/ssh/ssh_host_ed25519_key.pub
256 SHA256:8cnPrinC49ZHc+/9Ai5pV+1JfZ4WBRZhd3rDOsc2zlA root@halof (ED25519)
+--[ED25519 256]--+
|              +oo|
|             .+o.|
|        .    ..E.|
|         + .  +.o|
|        S +  + *o|
|          ooo Oo=|
|     . . . =o+.==|
|      = o =oo o=o|
|     o.o +o+..o.+|
+----[SHA256]-----+
```

Túneles de puertos SSH

OpenSSH cuenta con un mecanismo de reenvío muy potente por el que el tráfico de un puerto de origen se tuneliza -y encripta- a través de un proceso SSH que luego lo redirige a un puerto de un host de destino. Este mecanismo se conoce como *túnel de puertos* o *reenvío de puertos* y tiene importantes ventajas como las siguientes:

- Permite saltarse los cortafuegos para acceder a los puertos de los hosts remotos.
- Facilita el acceso desde el exterior a un host de su red privada.
- Proporciona encriptación para todo el intercambio de datos.

A grandes rasgos, podemos diferenciar entre túnel de puerto local y remoto.

Túnel de puerto local

Se define un puerto localmente para reenviar el tráfico al host de destino a través del proceso SSH que se encuentra en el medio. El proceso SSH puede ejecutarse en el host local o en un servidor remoto. Por ejemplo, si por alguna razón quisiera tunelizar una conexión a `www.gnu.org` a través de SSH usando el puerto `8585` en su máquina local, haría algo como esto:

```
carol@debian:~$ ssh -L 8585:www.gnu.org:80 debian
carol@debian's password:
Linux debian 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) x86_64

Los programas incluidos en el sistema Debian GNU/Linux son software libre:
(...)
Last login: Sun Jun 28 13:47:27 2020 from 127.0.0.1
```

La explicación es la siguiente: con el parámetro `-L`, especificamos el puerto local `8585` para conectar con el puerto *http* `80` en `www.gnu.org` utilizando el proceso SSH que se ejecuta en `debian`— nuestro *localhost*. Podríamos haber escrito `ssh -L 8585:www.gnu.org:80 localhost` con el mismo efecto. Si ahora utiliza un navegador web para ir a `http://localhost:8585`, será redirigido a `www.gnu.org`. Para la demostración, utilizaremos `lynx` (el clásico navegador web en modo texto):

```
carol@debian:~$ lynx http://localhost:8585
(...)
  * Back to Savannah Homepage
      * Not Logged in
      * Login
      * New User
      * This Page
      * Language
      * Clean Reload
      * Printer Version
      * Search
      * _
(...)
```

Si quisiera hacer exactamente lo mismo pero conectándose a través de un proceso SSH ejecutado en `halof`, habría procedido así:


```
carol@debian:~$ ssh -L 8585:www.gnu.org:80 -Nf ina@192.168.1.77
Enter passphrase for key '/home/carol/.ssh/id_ecdsa':
carol@debian:~$
carol@debian:~$ lynx http://localhost:8585
(...)
  * Back to Savannah Homepage
     * Not Logged in
     * Login
     * New User
     * This Page
     * Language
     * Clean Reload
     * Printer Version
     * Search
     * _
(...)
```

Es importante que anote tres detalles en el comando:

- Gracias a la opción `-N` no hemos entrado en `halof` sino que hemos hecho el reenvío de puertos.
- La opción `f` le indica a SSH que se ejecute en segundo plano.
- Especificamos el usuario `ina` para hacer el reenvío: `ina@192.168.1.77`

Túnel de puerto remoto

En el tunelado de puertos remotos (o reenvío inverso de puertos) el tráfico que llega a un puerto del servidor remoto es reenviado al proceso SSH que se ejecuta en su host local, y de ahí al puerto especificado en el servidor de destino (que también puede ser su máquina local). Por ejemplo, digamos que quiere que alguien de fuera de su red acceda al servidor web Apache que se ejecuta en su máquina local a través del puerto `8585` del servidor SSH que se ejecuta en `halof` (`192.168.1.77`). Usted procedería con el siguiente comando:

```
carol@debian:~$ ssh -R 8585:localhost:80 -Nf ina@192.168.1.77
Enter passphrase for key '/home/carol/.ssh/id_ecdsa':
carol@debian:~$
```

Ahora cualquiera que establezca una conexión con `halof` en el puerto `8585` verá la página de inicio por defecto de `Debian` Apache2:

```
carol@debian:~$ lynx 192.168.1.77:8585
(...)
                                                                 Apache2 Debian
Default Page: It works (p1 of 3)
   Debian Logo Apache2 Debian Default Page
   It works!

   This is the default welcome page used to test the correct operation of the
Apache2 server after
   installation on Debian systems. If you can read this page, it means that the
Apache HTTP server
   installed at this site is working properly. You should replace this file
(located at
   /var/www/html/index.html) before continuing to operate your HTTP server.
(...)
```

NOTE

Existe un tercer tipo de reenvío de puertos, más complejo, que queda fuera del ámbito de esta lección: el *reenvío dinámico de puertos*. En lugar de interactuar con un solo puerto, este tipo de reenvío utiliza varias comunicaciones TCP a través de un rango de puertos.

Túneles X11

Ahora que entiende los túneles de puertos, vamos a completar esta lección hablando del túnel X11 (también conocido como *X11forwarding*). A través de un túnel X11, el *X Window System* del host remoto es reenviado a su máquina local. Para ello, sólo tiene que pasar a `ssh` la opción `-X`:

```
carol@debian:~$ ssh -X ina@halof
...
```

Ahora puede lanzar una aplicación gráfica como el navegador web `firefox` con el siguiente resultado: la aplicación se ejecutará en el servidor remoto, pero su visualización se remitirá a su host local.

Si inicia una nueva sesión SSH con la opción `-x` en su lugar, el *X11forwarding* se desactivará. Intente iniciar `firefox` ahora y obtendrá un error como el siguiente:

```
carol@debian:~$ ssh -x ina@halof
carol@192.168.0.106's password:
(...)
ina@halof:~$ firefox

(firefox-esr:1779): Gtk-WARNING **: 18:45:45.603: Locale not supported by C
library.
    Using the fallback 'C' locale.
Error: no DISPLAY environment variable specified
```

NOTE Las tres directivas de configuración relacionadas con el reenvío de puertos locales, el reenvío de puertos remotos y el reenvío de X11 son `AllowTcpForwarding`, `GatewayPorts` y `X11Forwarding`, respectivamente. Para más información, escriba `man ssh_config` y/o `man sshd_config`.

Ejercicios guiados

1. Conectado como usuario `sonya` en su máquina cliente, realice las siguientes tareas SSH en el servidor remoto `halof`:

   ◦ Ejecute el comando para listar el contenido de `~/.ssh` como usuario `serena` en el host remoto; luego vuelva a su terminal local.

   ◦ Ingrese como usuario `serena` en el host remoto.

   ◦ Ingrese como usuario `sonya` en el host remoto.

   ◦ Borre todas las claves pertenecientes a `halof` de su archivo local `~/.ssh/known_hosts`.

   ◦ En su máquina cliente, cree un par de claves `ecdsa` de 256 bits.

   ◦ En su máquina cliente, cree un par de claves `ed25519` de 256 bits.

2. Ponga los siguientes pasos en el orden correcto para establecer una conexión SSH usando el agente de autenticación *SSH*:

   ◦ En el cliente, inicie un nuevo shell Bash para el *agente de autenticación* con `ssh-agent /bin/bash`.

   ◦ En el cliente, cree un par de claves con `ssh-keygen`.

   ◦ En el cliente, añada su clave privada a un área segura de la memoria con `ssh-add`.

   ◦ Añada la clave pública de su cliente al archivo `~/.ssh/authorized_keys` del usuario con el que quiere iniciar sesión en el host remoto.

   ◦ Si no existe ya, cree `~/.ssh` para el usuario con el que quiere iniciar sesión en el servidor.

   ◦ Conéctese al servidor remoto.

   El orden correcto es:

Paso 1:	
Paso 2:	
Paso 3:	
Paso 4:	
Paso 5:	
Paso 6:	

3. En cuanto al *port forwarding*, qué opción y directiva se utiliza para los siguientes tipos de túneles:

Tipo de túnel	**Opción**	**Directiva**
Local		
Remoto o Reverso		
X		

1. Suponga que escribe el comando `ssh -L 8888:localhost:80 -Nf ina@halof` en el terminal de su máquina cliente. Todavía en la máquina cliente, apunta un navegador a `http://localhost:8888`. ¿Qué obtendrá?

Ejercicios de exploración

1. Con respecto a las directivas de seguridad SSH:
   - Qué directiva se utiliza en `/etc/ssh/sshd_config` para habilitar los inicios de sesión `root`:
   - ¿Qué directiva usaría en `/etc/ssh/sshd_config` para especificar sólo una cuenta local para aceptar conexiones SSH:
2. Cuando se utiliza el mismo usuario tanto en el cliente como en el servidor, ¿qué comando `ssh` se puede utilizar para transferir la clave pública del cliente al servidor para poder iniciar sesión a través de la autenticación de clave pública?
3. Cree dos túneles de puertos locales en un solo comando reenviando los puertos locales no privilegiados 8080 y 8585 a través del servidor remoto `halof` a los sitios web `www.gnu.org` y `www.melpa.org`, respectivamente. Utilice el usuario `ina` en el servidor remoto y no olvide utilizar los interruptores `-Nf`:

Resumen

En esta lección hemos hablado de *OpenSSH 2*, que utiliza el protocolo *Secure Shell* para cifrar las comunicaciones entre el servidor y el cliente. Ha aprendido:

- Cómo iniciar sesión en un servidor remoto.
- Cómo ejecutar comandos de forma remota.
- Cómo crear pares de claves de autenticación.
- Cómo establecer inicios de sesión basados en claves.
- Cómo utilizar el *agente de autenticación* para mayor seguridad y comodidad.
- Los algoritmos de clave pública soportados por *OpenSSH*: `RSA`, `DSA`, `ecdsa`, `ed25519`.
- El papel de las claves de host *OpenSSH*.
- Cómo crear túneles de puertos: local, remoto y X.

El siguiente comando fue discutido en esta lección:

`ssh`
: Acceder o ejecutar comandos en una máquina remota.

`ssh-keygen`
: Generar, gestionar y convertir claves de autenticación.

`ssh-agent`
: Agente de autenticación OpenSSH.

`ssh-add`
: Añade identidades de clave privada al agente de autenticación.

Respuestas a los ejercicios guiados

1. Conectado como usuario `sonya` en su máquina cliente, realice las siguientes tareas SSH en el servidor remoto `halof`:

   - Ejecute el comando para listar el contenido de `~/.ssh` como usuario `serena` en el host remoto; luego vuelva a su terminal local.

   ```
   ssh serena@halof ls .ssh
   ```

   - Ingrese como usuario `serena` en el host remoto.

   ```
   ssh serena@halof
   ```

   - Ingrese como usuario `sonya` en el host remoto.

   ```
   ssh halof
   ```

   - Borre todas las claves pertenecientes a `halof` de su archivo local `~/.ssh/known_hosts`.

   ```
   ssh-keygen -R halof
   ```

   - En su máquina cliente, cree un par de claves `ecdsa` de 256 bits.

   ```
   ssh-keygen -t ecdsa -b 256
   ```

   - En su máquina cliente, cree un par de claves `ed25519` de 256 bits.

   ```
   ssh-keygen -t ed25519
   ```

2. Ponga los siguientes pasos en el orden correcto para establecer una conexión SSH usando el agente de autenticación *SSH*:

   - En el cliente, inicie un nuevo shell Bash para el *agente de autenticación* con `ssh-agent /bin/bash`.
   - En el cliente, cree un par de claves con `ssh-keygen`.

- En el cliente, añada su clave privada a un área segura de la memoria con `ssh-add`.
- Añada la clave pública de su cliente al archivo `~/.ssh/authorized_keys` del usuario con el que quiere iniciar sesión en el host remoto.
- Si no existe ya, cree `~/.ssh` para el usuario con el que quiere iniciar sesión en el servidor.
- Conéctese al servidor remoto.

  El orden correcto es:

Paso 1:	En el cliente, cree un par de claves utilizando `ssh-keygen`.
Paso 2:	Si no existe ya, cree `~/.ssh` para el usuario con el que quiere iniciar sesión en el servidor.
Paso 3:	Agregue la clave pública de su cliente al archivo `~/.ssh/authorized_keys` del usuario con el que quiere iniciar sesión en el host remoto.
Paso 4:	En el cliente, inicie un nuevo shell Bash para el *agente de autenticación* con `ssh-agent /bin/bash`.
Paso 5:	En el cliente, agregue su clave privada a una zona segura de la memoria con `ssh-add`.
Paso 6:	Conéctese con el servidor remoto.

3. En cuanto al *port forwarding*, qué opción y directiva se utiliza para los siguientes tipos de túneles:

Tipo de túnel	Opción	Directiva
Local	`-L`	`AllowTcpForwarding`
Remoto o Reverso	`-R`	`GatewayPorts`
X	`-X`	`X11Forwarding`

4. Suponga que escribe el comando `ssh -L 8888:localhost:80 -Nf ina@halof` en el terminal de su máquina cliente. Todavía en la máquina cliente, apunta un navegador a `http://localhost:8888`. ¿Qué obtendrá?

La página de inicio del servidor web de `halof`, como se entiende `localhost` desde la perspectiva del servidor.

Respuestas a los ejercicios de exploración

1. Con respecto a las directivas de seguridad SSH:
   - Qué directiva se utiliza en `/etc/ssh/sshd_config` para habilitar los inicios de sesión `root`:

     `PermitRootLogin`

   - ¿Qué directiva usaría en `/etc/ssh/sshd_config` para especificar sólo una cuenta local para aceptar conexiones SSH:

     `AllowUsers`

2. Cuando se utiliza el mismo usuario tanto en el cliente como en el servidor, ¿qué comando `ssh` se puede utilizar para transferir la clave pública del cliente al servidor para poder iniciar sesión a través de la autenticación de clave pública?

   `ssh-copy-id`

3. Cree dos túneles de puertos locales en un solo comando reenviando los puertos locales no privilegiados 8080 y 8585 a través del servidor remoto `halof` a los sitios web `www.gnu.org` y `www.melpa.org`, respectivamente. Utilice el usuario `ina` en el servidor remoto y no olvide utilizar los interruptores `-Nf`:

   `ssh -L 8080:www.gnu.org:80 -L 8585:www.melpa.org:80 -Nf ina@halof`



110.3 Lección 2

Versión:	5.0
Tema:	110 Seguridad
Objetivo:	110.3 Protección de datos mediante cifrado
Lección:	2 de 2

Introducción

En la lección anterior aprendimos a utilizar *OpenSSH* para cifrar las sesiones de inicio de sesión remotas, así como cualquier otro intercambio de información posterior. Puede haber otros escenarios en los que quiera cifrar archivos o correos electrónicos para que lleguen a su destinatario de forma segura y libre de miradas indiscretas. También puede necesitar firmar digitalmente esos archivos o mensajes para evitar que sean manipulados.

Una gran herramienta para este tipo de usos es el *GNU Privacy Guard* (también conocido como *GnuPG* o simplemente *GPG*), que es una implementación libre y de código abierto del sistema propietario *Pretty Good Privacy* (*PGP*). *GPG* utiliza el estándar *OpenPGP* definido por el *OpenPGP Working Group* del *Internet Engineering Task Force* (*IETF*) en el RFC 4880. En esta lección revisaremos los fundamentos del *GNU Privacy Guard*.

Configuración básica de GnuPG, uso y revocación

Al igual que con SSH, el mecanismo subyacente a GPG es el de la *criptografía asimétrica* o *criptografía de clave pública*. Un usuario genera un par de claves que se compone de una *clave privada* y una *clave pública*. Las claves están relacionadas matemáticamente de tal manera que lo que se cifra con una sólo puede ser descifrado por la otra. Para que la comunicación se realice con éxito, el usuario debe enviar su clave pública al destinatario.

Configuración y uso de GnuPG

El comando para trabajar con GPG es `gpg`. Puede pasarle una serie de opciones para realizar diferentes tareas. Empecemos por generar un par de claves como usuario `carol`. Para ello, utilizaremos el comando `gpg --gen-key`:

```
carol@debian:~$ gpg --gen-key
gpg (GnuPG) 2.2.12; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/carol/.gnupg' created
gpg: keybox '/home/carol/.gnupg/pubring.kbx' created
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name:

(...)
```

Después de informarle (entre otras cosas) que el directorio de configuración `~/.gnupg` y su llavero público `~/.gnugpg/pubring.kbx` han sido creados, `gpg` pasa a pedirle que proporcione su nombre real y su dirección de correo electrónico:

```
(...)
Real name: carol
Email address: carol@debian
You selected this USER-ID:
    "carol <carol@debian>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit?
```

Si está de acuerdo con el `USER-ID` resultante y pulsa `O`, se le pedirá una frase de contraseña (se recomienda que tenga suficiente complejidad):

```
┌──────────────────────────────────────────────────────┐
│ Please enter the passphrase to                       │
│ protect your new key                                 │
│                                                      │
│ Passphrase:  _                                       
(...)
```

Se mostrarán algunos mensajes finales que le informarán sobre la creación de otros archivos, así como de las propias claves, y entonces habrá terminado el proceso de generación de claves:

```
(...)
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/carol/.gnupg/trustdb.gpg: trustdb created
gpg: key 19BBEFD16813034E marked as ultimately trusted
gpg: directory '/home/carol/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/carol/.gnupg/openpgp-
revocs.d/D18FA0021F644CDAF57FD0F919BBEFD16813034E.rev'
public and secret key created and signed.

pub   rsa3072 2020-07-03 [SC] [expires: 2022-07-03]
      D18FA0021F644CDAF57FD0F919BBEFD16813034E
uid                      carol <carol@debian>
sub   rsa3072 2020-07-03 [E] [expires: 2022-07-03]
```

Ahora puede ver lo que hay dentro del directorio `~/.gnupg` (el directorio de configuración de GPG):

```
carol@debian:~/.gnupg$ ls -l
total 16
drwx------ 2 carol carol 4096 Jul  3 23:34 openpgp-revocs.d
drwx------ 2 carol carol 4096 Jul  3 23:34 private-keys-v1.d
-rw-r--r-- 1 carol carol 1962 Jul  3 23:34 pubring.kbx
-rw------- 1 carol carol 1240 Jul  3 23:34 trustdb.gpg
```

Expliquemos el uso de cada archivo:

`opengp-revocs.d`

Aquí se guarda el certificado de revocación que se creó junto con el par de claves. Los permisos de este directorio son bastante restrictivos, ya que cualquiera que tenga acceso al certificado podría revocar la clave (más información sobre la revocación de claves en la siguiente subsección).

`private-keys-v1.d`

Este es el directorio que guarda sus claves privadas, por lo que los permisos son restrictivos.

`pubring.kbx`

Este es su llavero público. Almacena sus propias claves públicas, así como cualquier otra importada.

`trustdb.gpg`

La base de datos de confianza. Esto tiene que ver con el concepto de *Web of Trust* (que está fuera del alcance de esta lección).

NOTE La llegada de *GnuPG 2.1* trajo consigo algunos cambios significativos, como la desaparición de los archivos `secring.gpg` y `pubring.gpg` en favor de `private-keys-v1.d` y `pubring.kbx`, respectivamente.

Una vez creado su par de claves, puede ver sus claves públicas con `gpg --list-keys` que mostrará el contenido de su llavero público:

```
carol@debian:~/.gnupg$ gpg --list-keys
/home/carol/.gnupg/pubring.kbx
------------------------------
pub   rsa3072 2020-07-03 [SC] [expires: 2022-07-03]
      D18FA0021F644CDAF57FD0F919BBEFD16813034E
uid           [ultimate] carol <carol@debian>
sub   rsa3072 2020-07-03 [E] [expires: 2022-07-03]
```

La cadena hexadecimal `D18FA0021F644CDAF57FD0F919BBEFD16813034E` es su *huella digital de clave pública*.

NOTE Además del `USER-ID` (`carol` en el ejemplo), también existe el `KEY-ID`. El mismo consiste en los últimos 8 dígitos hexadecimales de la huella digital de tu clave pública (`6813 034E`). Puede comprobar su huella digital con el comando `gpg --fingerprint` *`USER-ID`*.

Distribución y revocación de claves

Ahora que tiene su clave pública, debe guardarla (es decir, *exportarla*) en un archivo para ponerla a disposición de sus futuros destinatarios. Ellos podrán utilizarla para encriptar archivos o mensajes destinados a usted (como es el único que posee la clave privada, también será el único capaz de desencriptarlos y leerlos). Del mismo modo, tus destinatarios también la utilizarán para descifrar y verificar sus mensajes/archivos cifrados o firmados. El comando a utilizar es `gpg --export` seguido del `USER-ID` y una redirección al nombre del archivo de salida que elija:

```
carol@debian:~/.gnupg$ gpg --export carol > carol.pub.key
carol@debian:~/.gnupg$ ls
carol.pub.key  openpgp-revocs.d  private-keys-v1.d  pubring.kbx  trustdb.gpg
```

NOTE Pasando la opción `-a` o `--armor` a `gpg --export` (por ejemplo: `gpg --export --armor carol > carol.pub.key`) se creará una salida blindada ASCII (en lugar del formato binario OpenPGP por defecto) que puede ser enviada por correo electrónico de forma segura.

Como ya hemos dicho, ahora debe enviar su archivo de clave pública (`carol.pub.key`) al destinatario con el que quiere intercambiar información. Por ejemplo, enviemos el archivo de clave pública a `ina` en el servidor remoto `halof` utilizando `scp`(*secure copy*):

```
carol@debian:~/.gnupg$ scp carol.pub.key ina@halof:/home/ina/
Enter passphrase for key '/home/carol/.ssh/id_ecdsa':
carol.pub.key
100% 1740   775.8KB/s   00:00
carol@debian:~/.gnupg$
```

`ina` está ahora en posesión de `carol.pub.key`. La utilizará para encriptar un archivo y enviarlo a `carol` en la siguiente sección.

NOTE Otro medio de distribución de claves públicas es mediante el uso de *servidores de claves*: sube su clave pública al servidor con el comando `gpg --keyserver` *`keyserver-name`* `--send-keys` *`KEY-ID`* y otros usuarios las obtendrán (es decir, las *importarán*) con `gpg --keyserver` *`keyserver-name`* `--recv-keys` *`KEY-ID`*.

Terminemos esta sección hablando de la revocación de claves. La revocación de claves debe ser utilizada cuando sus claves privadas han sido comprometidas o retiradas. El primer paso es crear un certificado de revocación pasando a `gpg` la opción `--gen-revoke` seguida del `USER-ID`. Puede preceder

a `--gen-revoke` con la opción `--output` seguida de una especificación de nombre de archivo de destino para guardar el certificado resultante en un archivo (en lugar de imprimirlo en la pantalla del terminal). Los mensajes de salida a lo largo del proceso de revocación son bastante autoexplicativos:

```
sonya@debian:~/.gnupg$ gpg --output revocation_file.asc --gen-revoke sonya

sec  rsa3072/0989EB7E7F9F2066 2020-07-03 sonya <sonya@debian>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 1
Enter an optional description; end it with an empty line:
> My laptop was stolen.
>
Reason for revocation: Key has been compromised
My laptop was stolen.
Is this okay? (y/N) y
ASCII armored output forced.
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data and make it available to others!
```

El certificado de revocación se ha guardado en el archivo `evocation_file.asc` (`asc` para el formato ASCII):

```
sonya@debian:~/.gnupg$ ls
openpgp-revocs.d  private-keys-v1.d  pubring.kbx  revocation_file.asc  trustdb.gpg
sonya@debian:~/.gnupg$ cat revocation_file.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: This is a revocation certificate

iQHDBCABCgAtFiEEiIVjfDnnpieFi0wvnlcN6yLCeHEFAl8ASx4PHQJzdG9sZW4g
bGFwdG9wAAoJEJ5XDesiwnhxT9YMAKkjQiMpo9Uyiy9hyvukPPSrLcmtAGLk4pKS
pLZfzA5kxa+HPQwBglAEvfNRR6VMxqXUgUGYC/IAyQQM62oNAcY2PCPrxyJNgVF7
8l4mMZKvW++5ikjZwyg6WWV0+w6oroeo9qruJFjcu752p4T+9gsHVa2r+KRqcPQe
aZ65sAvsBJlcsUDZqfWUXg2kQp9mNPCdQuqvDaKRgNCHA1zbzNFzXWVd2X5RgFo5
nY+tUP8ZQA9DTQPBLPcggICmfLopMPZYB2bft5geb2mMi2oNpf9CNPdQkdccimNV
aRjqdUP9C89PwTafBQkQiONlsR/dWTFcqprG5KOWQPA7xjeMV8wretdEgsyTxqHp
v1iRzwjshiJCKBXXvz7wSmQrJ4OfiMDHeS4ipR0AYdO8QCzmOzmcFQKikGSHGMy1
z/YRlttd6NZIKjf1TD0nTrFnRvPdsZOlKYSArbfqNrHRBQkgirOD4JPI1tYKTffq
iOeZFx25K+fj2+0AJjvrbe4HDo5m+Q==
=umI8
-----END PGP PUBLIC KEY BLOCK-----
```

Para revocar efectivamente su clave privada, ahora necesita fusionar el certificado con la clave, lo que se hace importando el archivo del certificado de revocación a su llavero:

```
sonya@debian:~/.gnupg$ gpg --import revocation_file.asc
gpg: key 9E570DEB22C27871: "sonya <sonya@debian>" revocation certificate imported
gpg: Total number processed: 1
gpg:    new key revocations: 1
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2022-07-04
```

Anote sus claves ahora y se le informará sobre su clave revocada:

```
sonya@debian:~/.gnupg$ gpg --list-keys
/home/sonya/.gnupg/pubring.kbx
pub   rsa3072 2020-07-04 [SC] [revoked: 2020-07-04]
      8885637C39E7A627858B4C2F9E570DEB22C27871
uid           [ revoked] sonya <sonya@debian>
```

Por último, pero no por ello menos importante, asegúrese de que la clave revocada esté disponible para cualquier parte que tenga claves públicas asociadas a ella (incluidos los servidores de claves).

Usar GPG para cifrar, descifrar, firmar y verificar archivos

En la sección anterior, `carol` envió su clave pública a `ina`. La usaremos ahora para discutir cómo GPG puede encriptar, desencriptar, firmar y verificar archivos.

Cifrado y descifrado de archivos

En primer lugar, `ina` debe importar la clave pública de `carol` (`carol.pub.key`) a su llavero para poder empezar a trabajar con ella:

```
ina@halof:~> gpg --import carol.pub.key
gpg: /home/ina/.gnupg/trustdb.gpg: trustdb created
gpg: key 19BBEFD16813034E: public key "carol <carol@debian>" imported
gpg: Total number processed: 1
gpg:               imported: 1
ina@halof:~> gpg --list-keys
/home/ina/.gnupg/pubring.kbx
----------------------------
pub   rsa3072 2020-07-03 [SC] [expires: 2022-07-03]
      D18FA0021F644CDAF57FD0F919BBEFD16813034E
uid           [ unknown] carol <carol@debian>
sub   rsa3072 2020-07-03 [E] [expires: 2022-07-03]
```

A continuación creará un archivo escribiendo algún texto en él y luego lo encriptará usando `gpg` (como no firmó la clave de `carol`, se le preguntará explícitamente si quiere usar esa clave):

```
ina@halof:~> echo "This is the message ..." > unencrypted-message
ina@halof:~> gpg --output encrypted-message --recipient carol --armor --encrypt
unencrypted-message
gpg: 0227347CC92A5CB1: There is no assurance this key belongs to the named user
sub  rsa3072/0227347CC92A5CB1 2020-07-03 carol <carol@debian>
 Primary key fingerprint: D18F A002 1F64 4CDA F57F  D0F9 19BB EFD1 6813 034E
      Subkey fingerprint: 9D89 1BF9 39A4 C130 E44B  1135 0227 347C C92A 5CB1

It is NOT certain that the key belongs to the person named
in the user ID.  If you really know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```

Desglosemos el comando `gpg`:

`--output encrypted-message`

Especificación del nombre del archivo para la versión encriptada del archivo original (`mensaje-encriptado` en el ejemplo).

`--recipient carol`

Especificación del `ID de usuario` del destinatario (`carol` en nuestro ejemplo). Si no se proporciona, GnuPG lo pedirá (a menos que se especifique `--default-recipient`).

`--armor`

Esta opción produce una salida blindada ASCII, que puede copiarse en un correo electrónico.

`--encrypt unencrypted-message`

Especificación del nombre del archivo original a cifrar.

Ahora puede enviar el mensaje encriptado a `carol` en `debian` usando `scp`:

```
ina@halof:~> scp encrypted-message carol@debian:/home/carol/
carol@debian's password:
encrypted-message                                                           100%
736     1.8MB/s   00:00
```

Si ahora se conecta como `carol` e intenta leer el `mensaje encriptado`, confirmará que en realidad está encriptado y, por tanto, es ilegible:

```
carol@debian:~$ cat encrypted-message
-----BEGIN PGP MESSAGE-----

hQGMAwInNHzJKlyxAQv/brJ8Ubs/xya35sbv6kdRKm1C7ONLxL3OueWA4mCs0Y/P
GBna6ZEUCrMEgl/rCyByj3Yq74kuiTmzxAIRUDdvHfj0TtrOWjVAqIn/fPSfMkjk
dTxKo1i55tLJ+sj17dGMZDcNBinBTP4U1atuN71A5w7vH+XpcesRcFQLKiSOmYTt
F7SN3/5x5J6io4ISn+b0KbJgiJNNx+Ne/ub4Uzk4NlK7tmBklyC1VRualtxcG7R9
1klBPYSld6fTdDwT1Y4MofpyILAiGMZvUR1RXauEKf7OIzwC5gWU+UQPSgeCdKQu
X7QL0ZIBS0Ug2XKrO1k93lmDjf8PWsRIml6n/hNelaOBA3HMP0b6Ozv1gFeEsFvC
IxhUYPb+rfuNFTMEB7xIO94AAmWB9N4qknMxdDqNE8WhA728Plw6y8L2ngsplY15
MR4lIFDpljA/CcVh4BXVe9j0TdFWDUkrFMfaIfcPQwKLXEYJp19XYIaaEazkOs5D
W4pENN0YOcX0KWyAYX6r0l8BF0rq/HMenQwqAVXMG3s8ATuUOeqjBbR1x1qCvRQP
CR/3V73aQwc2j5ioQmhWYpqxiro0yKX2Ar/E6rZyJtJYrq+CUk8O3JoBaudknNFj
pwuRwF1amwnSZ/MZ/9kMKQ==
=g1jw
-----END PGP MESSAGE-----
```

Sin embargo, como está en posesión de la clave privada, puede descifrar fácilmente el mensaje pasando a `gpg` la opción `--decrypt` seguida de la ruta al archivo cifrado (se requerirá la frase de contraseña de la clave privada):

```
carol@debian:~$ gpg --decrypt encrypted-message
gpg: encrypted with 3072-bit RSA key, ID 0227347CC92A5CB1, created 2020-07-03
      "carol <carol@debian>"
This is the message ...
```

También puede especificar la opción `--output` para guardar el mensaje en un nuevo archivo sin cifrar:

```
carol@debian:~$ gpg --output unencrypted-message --decrypt encrypted-message
gpg: encrypted with 3072-bit RSA key, ID 0227347CC92A5CB1, created 2020-07-03
      "carol <carol@debian>"
carol@debian:~$ cat unencrypted-message
This is the message ...
```

Firma y verificación de archivos

Además de encriptar, GPG también puede utilizarse para firmar archivos. La opción `--sign` es relevante aquí. Comencemos creando un nuevo mensaje (`message`) y firmándolo con la opción `--sign` (se requerirá la frase de contraseña de su clave privada):

```
carol@debian:~$ echo "This is the message to sign ..." > message
carol@debian:~$ gpg --output message.sig --sign message
(...)
```

Desglose del comando `gpg`:

`--output message`

Especificación del nombre de la versión firmada del archivo original (`mensaje.sig` en nuestro ejemplo).

`--sign message`

Ruta de acceso al archivo original.

NOTE Con `--sign` se comprime el documento y se firma. La salida está en formato binario.

A continuación transferiremos el archivo a `ina` en `halof` usando `scp mensaje.sig ina@halof:/home/ina` . De vuelta como `ina` en `halof`, ahora puede verificarla usando la opción `--verify`:

```
ina@halof:~> gpg --verify message.sig
gpg: Signature made Sat 04 jul 2020 14:34:41 CEST
gpg:                using RSA key D18FA0021F644CDAF57FD0F919BBEFD16813034E
gpg: Good signature from "carol <carol@debian>" [unknown]
(...)
```

Si también quiere leer el archivo, tiene que desencriptarlo a un nuevo archivo (`mensaje` en nuestro caso) usando la opción `--output`:

```
ina@halof:~> gpg --output message --decrypt message.sig
gpg: Signature made Sat 04 jul 2020 14:34:41 CEST
gpg:                using RSA key D18FA0021F644CDAF57FD0F919BBEFD16813034E
gpg: Good signature from "carol <carol@debian>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: D18F A002 1F64 4CDA F57F  D0F9 19BB EFD1 6813 034E
ina@halof:~> cat message
This is the message to sign ...
```

Agente GPG

Completaremos esta lección tocando brevemente el tema de `gpg-agent`. El `gpg-agent` es el demonio que gestiona las claves privadas para GPG (se inicia a petición de `gpg`). Para ver un resumen de las opciones más útiles, ejecute `gpg-agent --help` o `gpg-agent -h`:


```
carol@debian:~$ gpg-agent --help
gpg-agent (GnuPG) 2.2.4
libgcrypt 1.8.1
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Syntax: gpg-agent [options] [command [args]]
Secret key management for GnuPG

Options:

     --daemon                            run in daemon mode (background)
     --server                            run in server mode (foreground)
     --supervised                        run in supervised mode
 -v, --verbose                           verbose
 -q, --quiet                             be somewhat more quiet
 -s, --sh                                sh-style command output
 -c, --csh                               csh-style command output
(...)
```

NOTE Para más información, consulte la página man de `gpg-agent`.

Ejercicios guiados

1. Complete la tabla proporcionando el nombre de archivo correcto:

Descripción	**Archivo**
Base de datos de confianza	
Directorio de certificados de revocación	
Directorio de claves privadas	
Lector de claves públicas	

2. Responda a las siguientes preguntas:

   - ¿Qué tipo de criptografía utiliza *GnuPG*?

   - ¿Cuáles son los dos componentes principales de la criptografía de clave pública?

   - ¿Cuál es el `KEY-ID` de la huella digital de la clave pública `07A6 5898 2D3A F3DD 43E3 DA95 1F3F 3147 FA7F 54C7`?

   - ¿Qué método se utiliza para distribuir las claves públicas a nivel global?

3. Ponga los siguientes pasos en el orden correcto con respecto a la revocación de la clave privada:

   - Poner la clave revocada a disposición de sus corresponsales.
   - Crear un certificado de revocación.
   - Importar el certificado de revocación a su llavero.

   El orden correcto es:

Paso 1:	
Paso 2:	
Paso 3:	

4. En cuanto al cifrado de archivos, ¿qué implica la opción `--armor` en el comando `gpg --output encrypted-message --recipient carol --armor --encrypt unencrypted-message`?

Ejercicios de exploración

1. La mayoría de las opciones `gpg` tienen una versión larga y otra corta. Complete la tabla con la versión corta correspondiente:

Versión larga	Versión corta
`--armor`	
`--output`	
`--recipient`	
`--decrypt`	
`--encrypt`	
`--sign`	

2. Responda a las siguientes preguntas sobre la exportación de claves:
   - ¿Qué comando utilizaría para exportar todas sus claves públicas a un archivo llamado `all.key`?
   - ¿Qué comando utilizaría para exportar todas sus claves privadas a un archivo llamado `all_private.key`?

3. ¿Qué opción de `gpg` permite llevar a cabo la mayoría de las tareas relacionadas con la gestión de llaves presentando un menú?

4. ¿Qué opción de `gpg` permite realizar una firma en texto claro?

Resumen

Esta lección ha cubierto *GNU Privacy Guard*, una excelente opción para cifrar/descifrar y firmar/verificar digitalmente los archivos. Usted aprendió:

- Cómo generar un par de claves.
- Cómo listar las llaves en su llavero.
- El contenido del directorio `~/.gnupg`.
- Qué son `USER-ID` y `KEY-ID`.
- Cómo distribuir las claves públicas a sus corresponsales.
- Cómo dividir globalmente las claves públicas a través de servidores de claves.
- Cómo revocar las claves privadas.
- Cómo encriptar y desencriptar archivos.
- Cómo firmar y verificar los archivos.
- Los fundamentos del *GPG-Agent*.

El siguiente comando fue discutido en esta lección:

`gpg`

Herramienta de cifrado y firma *OpenPGP*.

Respuestas a los ejercicios guiados

1. Complete la tabla proporcionando el nombre de archivo correcto:

Descripción	**Archivo**
Base de datos de confianza	`trustdb.gpg`
Directorio de certificados de revocación	`opengp-revocs.d`
Directorio de claves privadas	`private-keys-v1.d`
Lector de claves públicas	`pubring.kbx`

2. Responda a las siguientes preguntas:
   - ¿Qué tipo de criptografía utiliza *GnuPG*?

     Criptografía de clave pública o asimétrica.

   - ¿Cuáles son los dos componentes principales de la criptografía de clave pública?

     Las claves pública y privada.

   - ¿Cuál es el `KEY-ID` de la huella digital de la clave pública `07A6 5898 2D3A F3DD 43E3 DA95 1F3F 3147 FA7F 54C7`?

     `FA7F 54C7`

   - ¿Qué método se utiliza para distribuir las claves públicas a nivel global?

     Key servers.

3. Ponga los siguientes pasos en el orden correcto con respecto a la revocación de la clave privada:
   - Poner la clave revocada a disposición de sus corresponsales
   - Crear un certificado de revocación
   - Importar el certificado de revocación a su llavero

   El orden correcto es:

Paso 1:	Crear un certificado de revocación
Paso 2:	Importar el certificado de revocación a su llavero

Paso 3:	Poner la clave revocada a disposición de sus corresponsales

4. En cuanto al cifrado de archivos, ¿qué implica la opción `--armor` en el comando `gpg --output encrypted-message --recipient carol --armor --encrypt unencrypted-message`?

   Produce una salida blindada ASCII, que permite copiar el archivo cifrado existente resultante en un correo electrónico.

Respuestas a los ejercicios de exploración

1. La mayoría de las opciones `gpg` tienen una versión larga y otra corta. Complete la tabla con la versión corta correspondiente:

Versión larga	Versión corta
`--armor`	`-a`
`--output`	`-o`
`--recipient`	`-r`
`--decrypt`	`-d`
`--encrypt`	`-e`
`--sign`	`-s`

2. Responda a las siguientes preguntas sobre la exportación de claves:
   - ¿Qué comando utilizaría para exportar todas sus claves públicas a un archivo llamado `all.key`?

     `gpg --export --output all.key` o `gpg --export -o all.key`

   - ¿Qué comando utilizaría para exportar todas sus claves privadas a un archivo llamado `all_private.key`?

     `gpg --export-secret-keys --output all_private.key` o `gpg --export-secret-keys -o all_private.key` (`--export-secret-keys` puede ser sustituido por `--export-secret-subkeys` con un resultado ligeramente diferente — consulte `man pgp` para más información).

3. ¿Qué opción de `gpg` permite llevar a cabo la mayoría de las tareas relacionadas con la gestión de llaves presentando un menú?

   `--edit-key`

4. ¿Qué opción de `gpg` permite realizar una firma en texto claro?

   `--clearsign`

Pie de imprenta

PDF generado: 2022-04-21

LPI Learning Materials son una iniciativa del Linux Professional Institute (https://lpi.org). Los materiales y sus traducciones pueden encontrarse en https://learning.lpi.org.

Para preguntas y comentarios sobre esta edición, así como sobre todo el proyecto, escriba un correo electrónico a: learning@lpi.org.